Topic Modeling on Lyrics for Genre Classification

Niklas Smedemark-Margulies
Khoury College
Northeastern University
Boston, MA
smedemark-margulie.n@husky.neu.edu

Daniel Zeiberg
Khoury College
Northeastern University
Boston, MA
zeiberg.d@husky.neu.edu

Xiongyi Fred Zhang
Khoury College
Northeastern University
Boston, MA
zhang.xio@husky.neu.edu

Abstract

In this project, we evaluated the quality of features learned from topic modeling by analyzing the performance in downstream classification tasks. We worked with a public dataset of 380,000+ song lyrics [1] to classify songs by genre. We compared two recent and closely related neural topic models, ProdLDA [2], and SCHOLAR [3], with baseline feature sets from word2Vec [4], doc2Vec [5], tf-idf [6], and raw bag-of-words.

1 Methods

Recall that topic models seek to find interpretable latent structure in data. These models are commonly used in natural language processing, but have also been used in recommender-systems and bioinformatics [7]. We compared the utility of features learned from two neural topic models to those learned from baseline methods in downstream classification tasks. We used ProdLDA, both the supervised and unsupervised versions of SCHOLAR, and baselines of Word2Vec, Doc2Vec, TF-IDF, and raw Bag-of-words.

1.1 SCHOLAR

Due to the substantial similarity between ProdLDA and SCHOLAR, we focus here on a terse description of SCHOLAR [3]. SCHOLAR is a neural topic model that optionally incorporates metadata and task labels in the topic model training. Including covariates allows one to incorporate structured knowledge in the topic modeling. In particular, supervised topic modeling allows one to learn topics that are useful in downstream classification tasks. One trains SCHOLAR by maximizing the Evidence Based Lower Bound (ELBO):

$$\mathcal{L} = \mathbb{E}_{q_\phi(r_i|w_i,c_i,y_i)}\left[\sum_{j=1}^{N_i} log(p(w_{i,j}|w_i, c_i)\right] + \mathbb{E}_{q_\phi(r_i|w_i,c_i,y_i)}\left[log(p(y_i|w_i, c_i))\right] - D_{KL}[q_\phi(r_i|w_i, c_i, y_i)||p(r_i|\alpha)]$$

Which we will approximate using sampling as:

$$\mathcal{L} \approx \sum_{j=1}^{N_i} log(p(w_{i,j}|r_i^{(s)}, c_i)) + log(p(y_i|r_i^{(s)}, c_i)) - D_{KL}[q_\phi(r_i|w_i, c_i, y_i)||p(r_i|\alpha)]$$

To generate document bag-of-words and labels, SCHOLAR approximates reparametrized sampling from a Dirichlet distribution by sampling from a logistic-normal prior and applying the softmax function (see section 3.1 of SCHOLAR):

$$r_i \sim \mathcal{N}(r_i|\mu_0(\alpha), diag(\sigma_0^2(\alpha)))$$
$$\theta_i = softmax(r_i)$$

This latent representation can then be used to generate words for a document:

$$\eta_i = f_g(\theta_i, c_i)$$

for each word j in document i:

$$w_{i,j} \sim p(w_{i,j}|softmax(\eta_i))$$

Labels can additionally be sampled

$$y_i \sim p(y_i|f_y(\theta_i, c_i))$$

In instances where labels or covariates are not present, zero vectors can be passed in.

1.2 Word2Vec

Word2Vec is model that learns a latent representation of words. There are two main variants of word2Vec: continuous bag-of-words (CBOW) and skip-gram. We defer to the default Gensim implementation of CBOW Word2Vec; the CBOW objective forces a neural net to predict a word given the neighboring words. In both cases, we can extract a low-dimensional vector embedding of the word of interest from a hidden layer in the network.

1.3 Doc2Vec

Doc2Vec is an extension of word2Vec models where document embeddings are learned jointly with word embeddings. While word2Vec learns a matrix W in which each column is the embedding of a word in the vocabulary, doc2Vec additionally learns a matrix D in which each column is an embedding of a document.

1.4 TF-IDF

Term frequency-inverse document frequency (TF-IDF) is an unsupervised model based on a document's bag-of-words representation. TF-IDF creates an embedding vector with size equal to that of the corpus vocabulary. The TF term intuitively weights a word as more important for that document if it shows up frequently in the document, while the IDF term intuitively weights a word as more important if it is a rare word in the corpus. We implement TF as the smoothed fraction of a document occupied by a certain word, and a smooth log-scaled IDF.

2 Experiment

2.1 Task

Our classification task was to predict the genre of a song based off the song's lyrics. We trained logistic regression models using features extracted from lyrics. We extracted features using supervised SCHOLAR, unsupervised SCHOLAR, ProdLDA, word2Vec, doc2Vec, and TF-IDF. We compared the utility of the supervised and unsupervised topic model in the downstream task of genre classification. Additionally, we evaluated the performance of supervised SCHOLAR in a semi-supervised setting.

2.2 Data and Preprocessing

For this task we use the MetroLyrics dataset, which contains the lyrics and genre labels for 380,000 songs. We removed songs without a specified genre or lyrics, leaving 10 genres and a total of 237,427

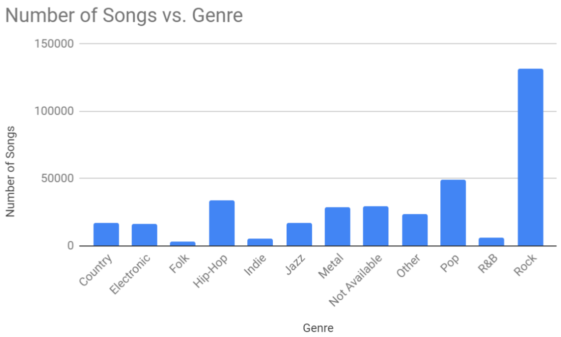


Figure 1: Genre distribution for full dataset prior to resampling

songs. For all models accepting bag-of-words input, we tokenized, stemmed, and used a vocabulary of the top 2000 words. Due to significant class imbalance, as seen in figure 1, after our train/test split, we resampled with replacement to 5,000 songs per genre, for a total of 50,000 songs. We used Gensim [8] default parameters to train Word2Vec, with window size 5 and minimum word count 2, and Doc2Vec, with minimum word count 2, on the full corpus for 100 epochs.

2.3 Topic Model Training

We based our work on code provided by the original authors of each neural model; we used Tensorflow to train ProdLDA, and PyTorch to train SCHOLAR. Due to resource limitations, we did not perform an exhaustive hyperparameter search, but focused first on the latent dimension (i.e. the number of topics or the word2vec/doc2vec target dimension). We used SGD with a batch size of 200 and 40 epochs to train each model, which we found was sufficient for convergence.

2.4 Downstream Classification

To extract a latent representation of each document, we simply ran a forward pass through our trained models without dropout and without batch normalization, and saved the posterior mean vector output from our encoder model. Due to time constraints, we ran a standard logistic regression without cross-validation or confidence intervals, though manual inspection shows that the results are consistent across runs to approximately 1-3% classification accuracy across the range of parameters used.

2.5 Semi-Supervised Learning, Reweighted Cost Function, and Hyperparameter Search

Based on our initial results in comparing classification performance, we wanted to further understand the key components necessary for learning effective features. We experimented with adding and tuning coefficients in our neural net cost function. The SCHOLAR model includes a classification layer as well as a classification term in its loss function. We added coefficients for the 3 terms in the loss function; reconstruction, KL-divergence from the prior, and classification loss, and performed a grid search over these hyperparameters, in the spirit of Beta-VAE[9]. We also used semi-supervised training; this explores a key real-world application scenario, and also helps make the relative importance of these loss function terms more apparent. To that end, we fixed the number of topics at 100 and performed a series of experiments while removing the labels from a varying percentage of our training data. During training, for each unlabeled example, only the reconstruction loss and the KL loss are computed. We simultaneously grid search across the coefficients semi-supervised learning, i.e. how much labeled data are we preserving.

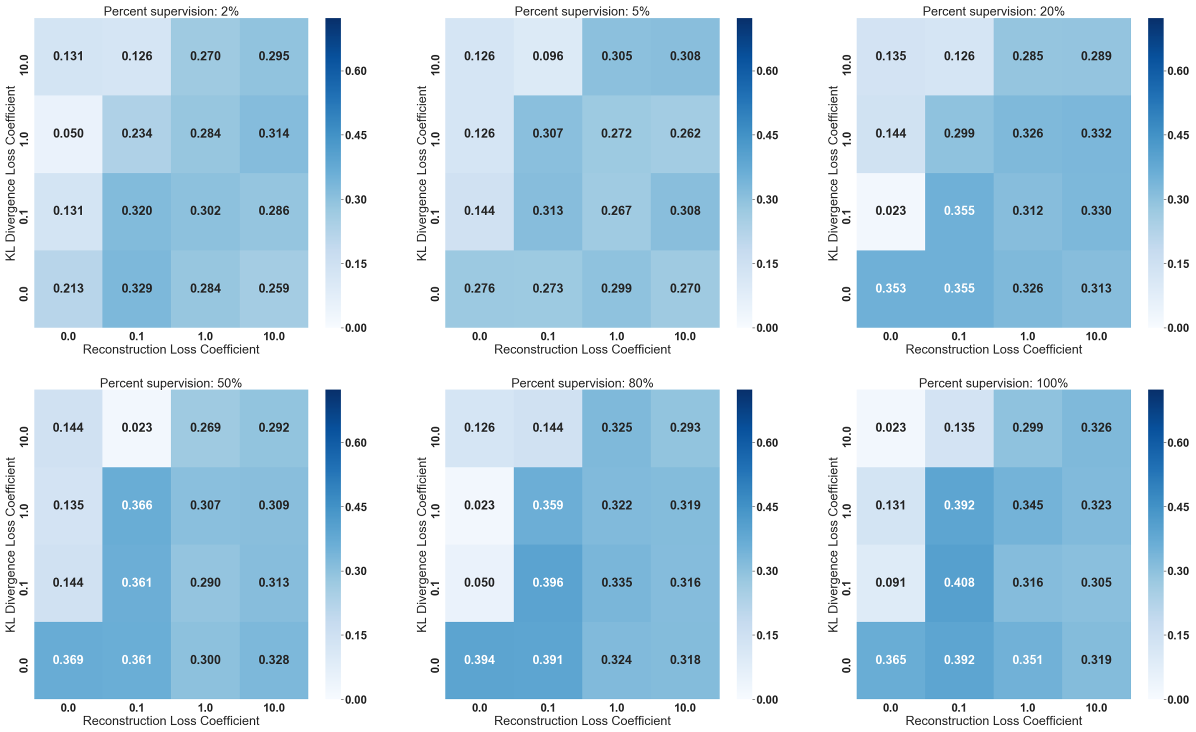


Figure 2: Semi-Supervised Scholar Test Accuracy while Varying Loss Function Coefficients with 100 topics

‘

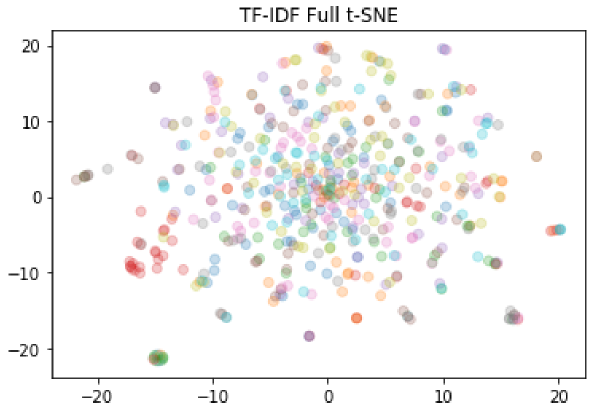


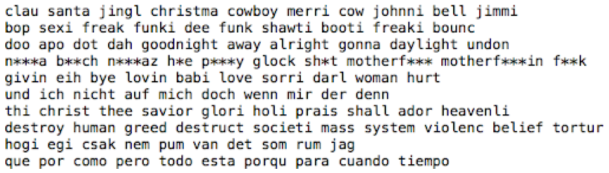
Figure 3: 300 dimensional TF-IDF t-SNE feature visualization. Color represents song genre

‘

3 Results

3.1 Visualizing Features

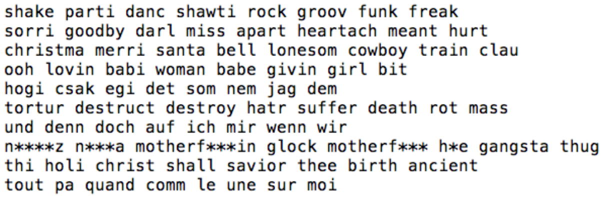
Figure 3 visualizes the features from 300 dimensional TF-IDF reduced to 2 dimensions using t-SNE. The genre of each song is encoded in the color channel. This visualization indicates that the genres are not well separated and shows the inherent difficulty in this classification task.



```
clau santa jingl christma cowboy merri cow johnni bell jimmi
bop sexi freak funki dee funk shawti booti freaki bounc
doo apo dot dah goodnight away alright gonna daylight undon
n***a b**ch n***az h*e p***y glock sh*t motherf*** motherf***in f**k
givin eih bye lovin babi love sorri darl woman hurt
und ich nicht auf mich doch wenn mir der denn
thi christ thee savior glori holi prais shall ador heavenli
destroy human greed destruct societi mass system violenc belief tortur
hogi egi csak nem pum van det som rum jag
que por como pero todo esta porqu para cuando tiempo
```

Figure 4: Top words in each topic for ProdLDA

‘



```
shake parti danc shawti rock groov funk freak
sorri goodby darl miss apart heartach meant hurt
christma merri santa bell lonesom cowboy train clau
ooh lovin babi woman babe givin girl bit
hogi csak egi det som nem jag dem
tortur destruct destroy hatr suffer death rot mass
und denn doch auf ich mir wenn wir
n****z n***a motherf***in glock motherf*** h*e gangsta thug
thi holi christ shall savior thee birth ancient
tout pa quand comm le une sur moi
```

Figure 5: Top words in each topic for Supervised SCHOLAR

‘

3.2 Qualitative Evaluation of Topic Models

To get a qualitative evaluation of how good are the topics we learned, and compare the topics learned by different topic models, we print the top words of each topic of the three different topic models; ProdLDA, unsupervised SCHOLAR and supervised SCHOLAR in figures 4, 5, 6. One can observe that profanities are restricted to a single topic in supervised SCHOLAR, while they appear in two topics in unsupervised scholar. This seems to indicate that supervised SCHOLAR better forms a topic for words that commonly appear in Hip-Hop lyrics.

3.3 Genre Prediction

In figure 7 we visualize the genre prediction accuracy for model using various latent representation dimensions. Notice that canonical NLP baseline models, TF-IDF and raw bag-of-words perform very well. Notice also that, while uniformly weighted supervised SCHOLAR does not outperform the simple baselines or unsupervised SCHOLAR (N = 100 topics, supervised SCHOLAR 33%, unsupervised SCHOLAR 35%), re-weighting the components of the loss function, such that classification loss is emphasized over KL divergence and reconstruction loss, results in improved classification accuracy, increasing from 33% to 40.8%, outperforming baseline models.

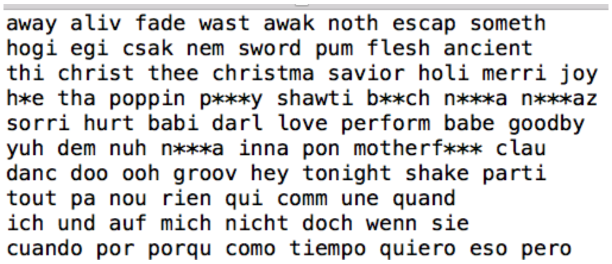
3.4 Semi-supervised learning and Loss Function Coefficient Modifications

We found several interesting trends in supervised SCHOLAR's performance as we sweep the hyperparameter grid. In the regime of plenty of labeled examples, we observed that the best performing model uses a cost function based only on the classification loss. In this scenario, our model essentially degenerates to a multi-layer perceptron, and this outcome may be seen as somewhat expected. Additionally, we see that increasing the relative weight on the KL term causes worse performance, which we may indicate that the latent structure of our data is not well captured by the logistic normal prior.

In low-label regime, we were interested to see that increasing the weight on the autoencoder loss terms does indeed improve model performance. Notice also that at the lowest range of percent supervision, there is increased variability in the model classification accuracy, which may also be, given the high noise in our data labels.

4 Conclusion

In this work we evaluate the utility of features learned from topic modeling in downstream document classification tasks. We find at low levels of supervision, the added reconstruction and KL loss terms improve predictive performance compared to the "MLP scenario". While we explored how



```
away aliv fade wast awak noth escap someth
hogi egi csak nem sword pum flesh ancient
thi christ thee christma savior holi merri joy
h*e tha poppin p***y shawti b**ch n***a n***az
sorri hurt babi darl love perform babe goodby
yuh dem nuh n***a inna pon motherf*** clau
danc doo ooh groov hey tonight shake parti
tout pa nou rien qui comm une quand
ich und auf mich nicht doch wenn sie
cuando por porqu como tiempo quiero eso pero
```

Figure 6: Top words in each topic for Unsupervised SCHOLAR
‘

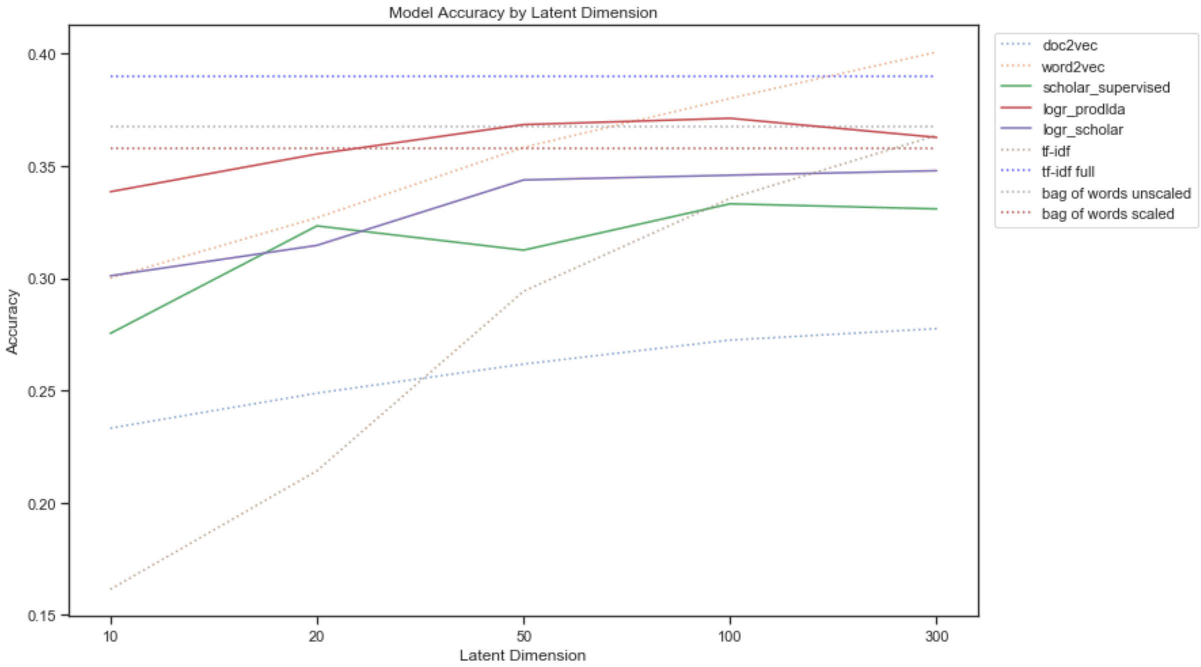


Figure 7: Genre Classification Accuracy by latent dimension
‘

robust these neural topic models are to semi-supervision, we could continue the hyperparameter search for other key parameters such as vocabulary size. Further work could also investigate the performance of baseline methods in low supervision scenarios to make a more complete comparison about the benefits of the VAE-based approach. Additionally, one could investigate alternative ways to incorporate the unlabeled data in the loss function, such as by attempting to minimize the entropy in predictions of unlabeled data.

Code for our experiments can be found on the CCS GitHub [1].

References

[1] GyanendraMishra. 380,000+ lyrics from metrolyrics, 2017. URL `https://www.kaggle.com/gyani95/380000-lyrics-from-metrolyrics`.

[2] Akash Srivastava and Charles Sutton. Autoencoding variational inference for topic models. *arXiv preprint arXiv:1703.01488*, 2017.

[1] https://github.ccs.neu.edu/dzeiberg/musicProject

[3] Dallas Card, Chenhao Tan, and Noah A Smith. Neural models for documents with metadata. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2031–2040, 2018.

[4] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[5] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196, 2014.

[6] Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21, 1972.

[7] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[8] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. `http://is.muni.cz/publication/884893/en`.

[9] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, volume 3, 2017.