Reddit Explorer: Topic Indexing and Page Recommendation

github.com/lmburke/reddit-analysis

Lee Burke

PNNL Interview

February 28th, 2018

Introduction

- Reddit.com had 250 million unique users in 2017
- Message board with posts organized by "subreddit"
- No easy way to find new subreddits (new communities!)
- Motivation: find new subreddits with similar interests to known communities

Examples of Subreddits

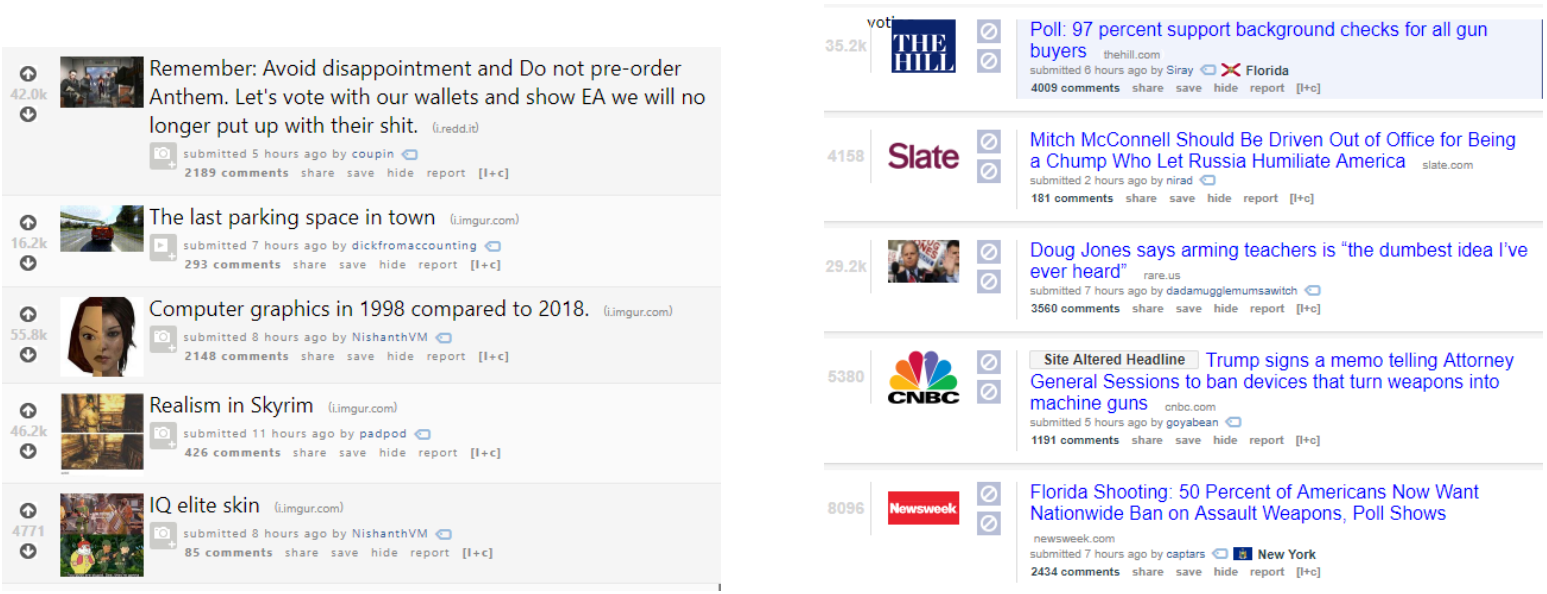


Figure 1: Posts on `/r/gaming` (left) and `/r/politics` (right)

Data Preparation

Figure 2: The most common words in `/r/gaming` (left) and `/r/politics` (right)

- Reddit API is rate-limited
- Google BigQuery monthly comment dump (October 2017)
- 1000 most popular subreddits, 2000 most upvoted comments
- Laptop-scale at 0.5 GB

Preprocessing

- Remove formatting, punctuation, and non-English characters
- Remove empty comments
- Lemmatize: combine conjugations into single word (NLTK)
- Vectorize: split on whitespace

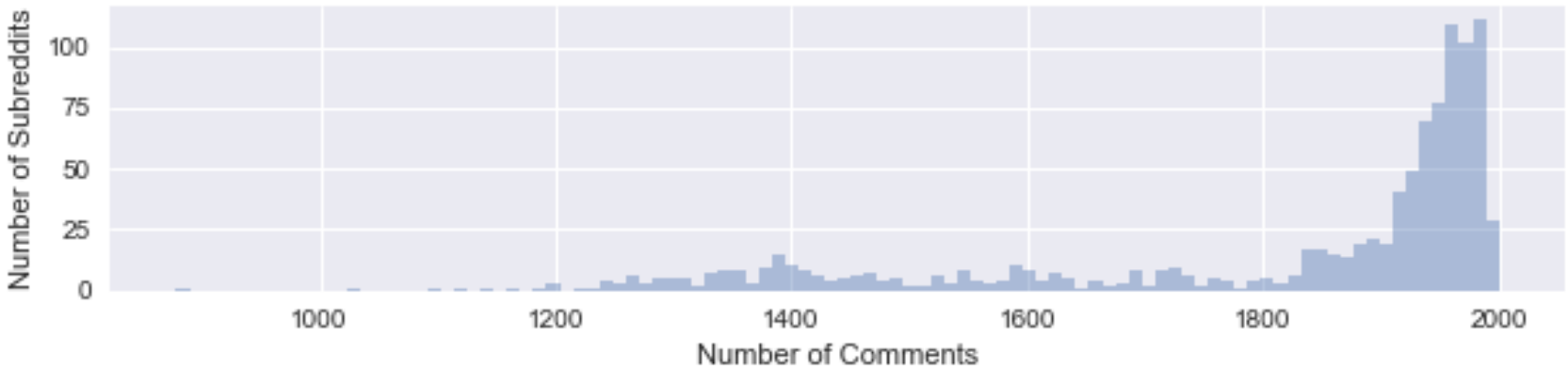


Figure 3: The number of subreddits with any given number of comments after preprocessing is shown.

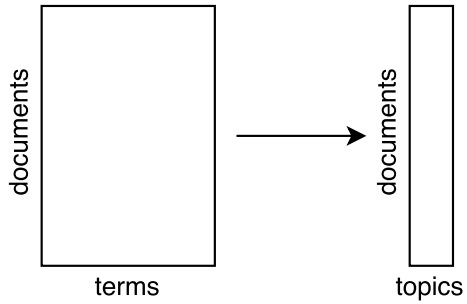
Vectorizing Raw Text

- Co-occurrence matrix: count of each word in each document
- Bag-of-words, n-grams
- Term frequency, inverse document frequency
- Minimum and maximum frequency

	document	frequency	word
Line 1	1		1
Line 2			1
Line 3	1	2	
Line 4		1	

Figure 4: A co-occurrence matrix of this slide

Vector Space Models



- High-dimensional, sparse vectors $\rightarrow$ smaller, dense vectors
- Basis of new space is a "topic", a mixture of words
- Each document is a mixture of topics
- Latent semantic analysis (LSA)
  - Singular value decomposition
  - Maximizes explained co-variance
  - Permits topics with negative weights

Statistical Models

- Probabilistic LSA (pLSA)
  - Model co-occurrence as multinomial mixture
  - Equivalent to a form of non-negative matrix factorization
- Latent Dirichlet Allocation (LDA)
  - Blei et al., 2003
  - Assume sparse Dirichlet priors
  - Few words per topic, few topics per document
  - Can be cast as tensor spectral decomposition
  - Usually use mini-batch expectation-maximization

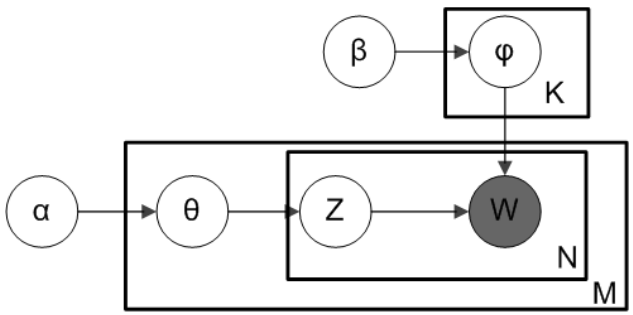


Figure 5: Plate diagram for the LDA model

Word Embedding Models

$$\text{Queen} - \text{Woman} + \text{Man} = \text{King}$$

- Capture the meaning of words in a vector space
- word2vec (Mikolov et al., 2013)
  - Dense, shallow neural network
  - Current word from its context (continuous BoW)
  - Nearby words from current word (skip-gram)

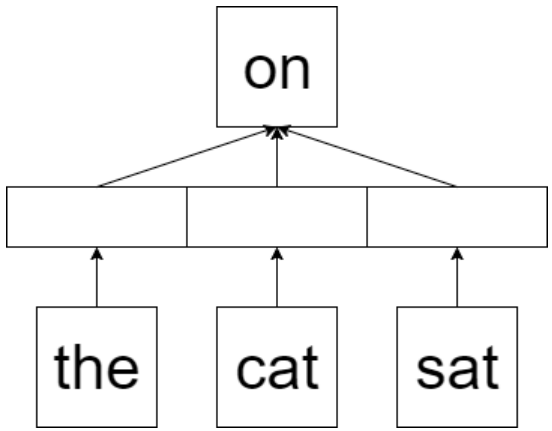


Figure 6: A framework for learning word vectors

Word Embedding Models: Documents

- How to capture the meaning of a sentence?
  - Weighted average of vectors
  - Expensive semantic parsing
- doc2vec (Le & Mikolov, 2014)
  - Embed document meaning in (different) vector space
  - Include document vector in word2vec framework

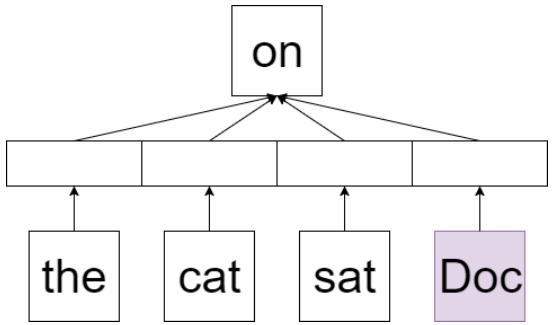


Figure 7: A framework for learning document vectors

Clustering & Recommendation

- "Find similar communities" from vector of document features
- Recommendation: return n results
  - Nearest Neighbors: determine nearest samples in feature vector space, according to some metric
  - Annoy: approximate NN using random projections and tree search
- Clustering (or segmentation): return all similar results
  - KMeans: find "prototype" mean value to minimize variance
  - Birch: build hierarchical tree of sub-clusters, then apply existing agglomerative clustering

Comparing Models

- Unsupervised problems hard to assess: no gold-standard labels
- Unsupervised metrics measure cohesion and separation
- Biased toward similar objectives: e.g., Silhouette Score

Truth: 0.39 ± 0.02 KMeans: 0.48 ± 0.02 Birch: 0.18 ± 0.02

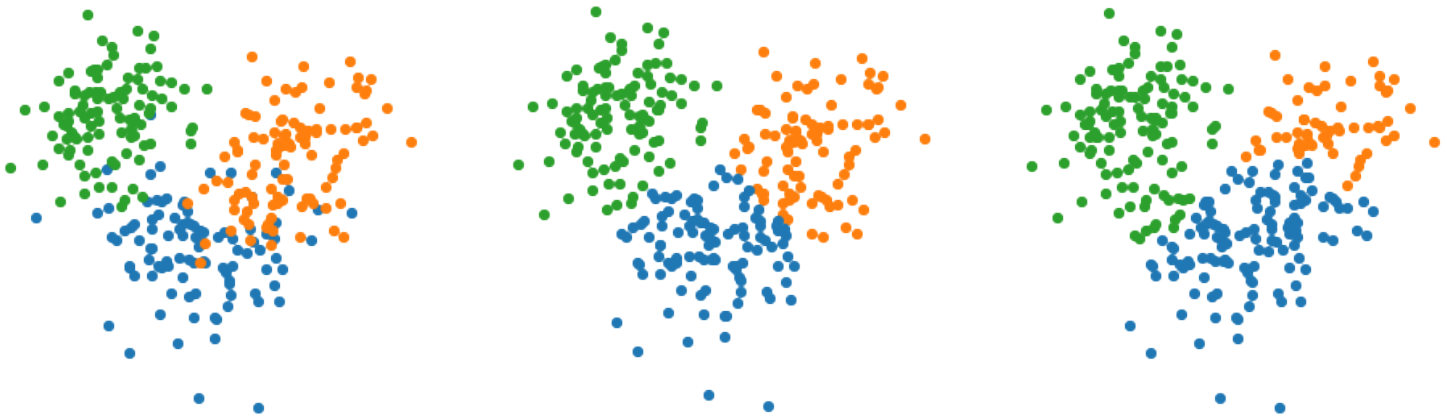


Figure 8: Artificial data: true labels (left), KMeans (center), Birch (right)

Classification

- Objective metrics: accuracy, precision, recall, F-β scores
- Comparing model performance
  - Hyperparameter tuning
  - Feature selection methods
- Cross-validated experiments are expensive: down-sample
  - LSA and KNN
  - LDA
  - doc2vec
- Select on final accuracy test
- Tune and select clustering methods

LSA and KNN

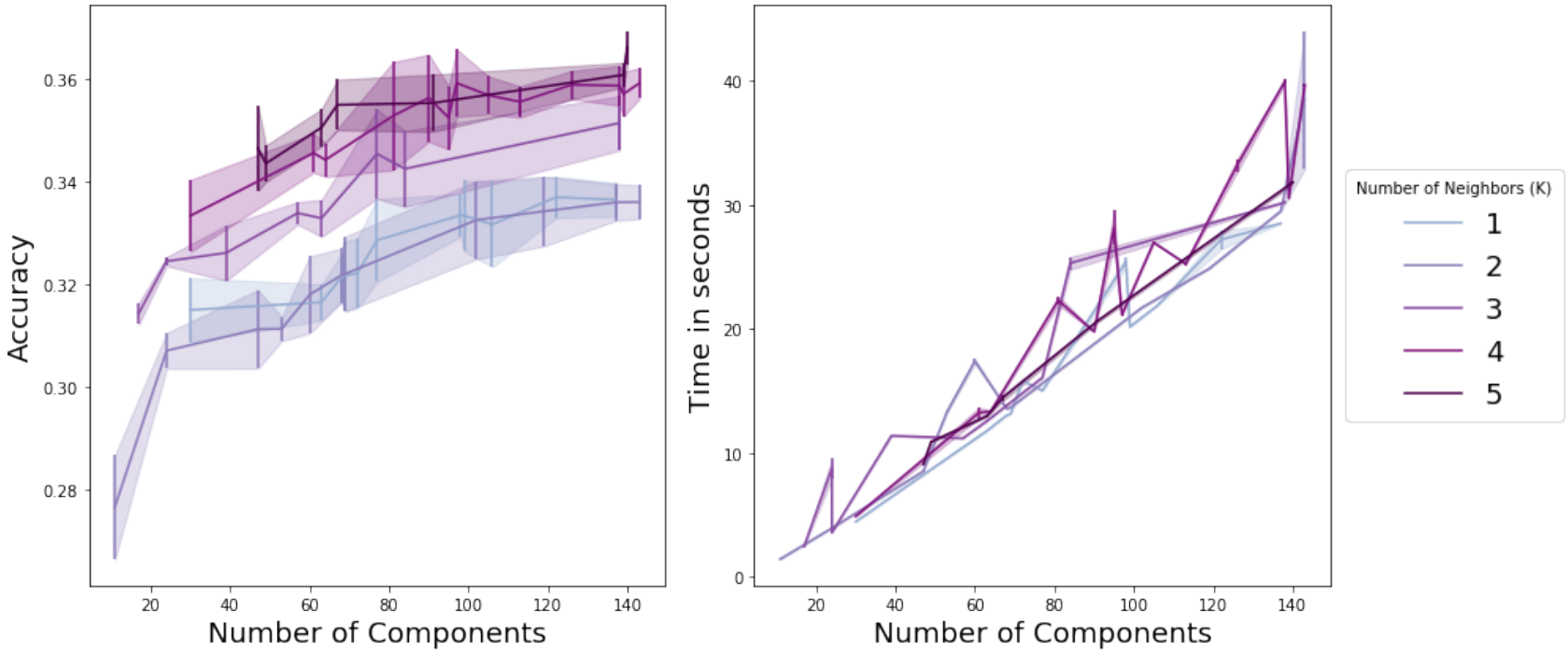


Figure 9: Accuracy and time are plotted against the number of LSA components for various K in KNN. Shaded regions represent one standard deviation in cross-validation.

LDA

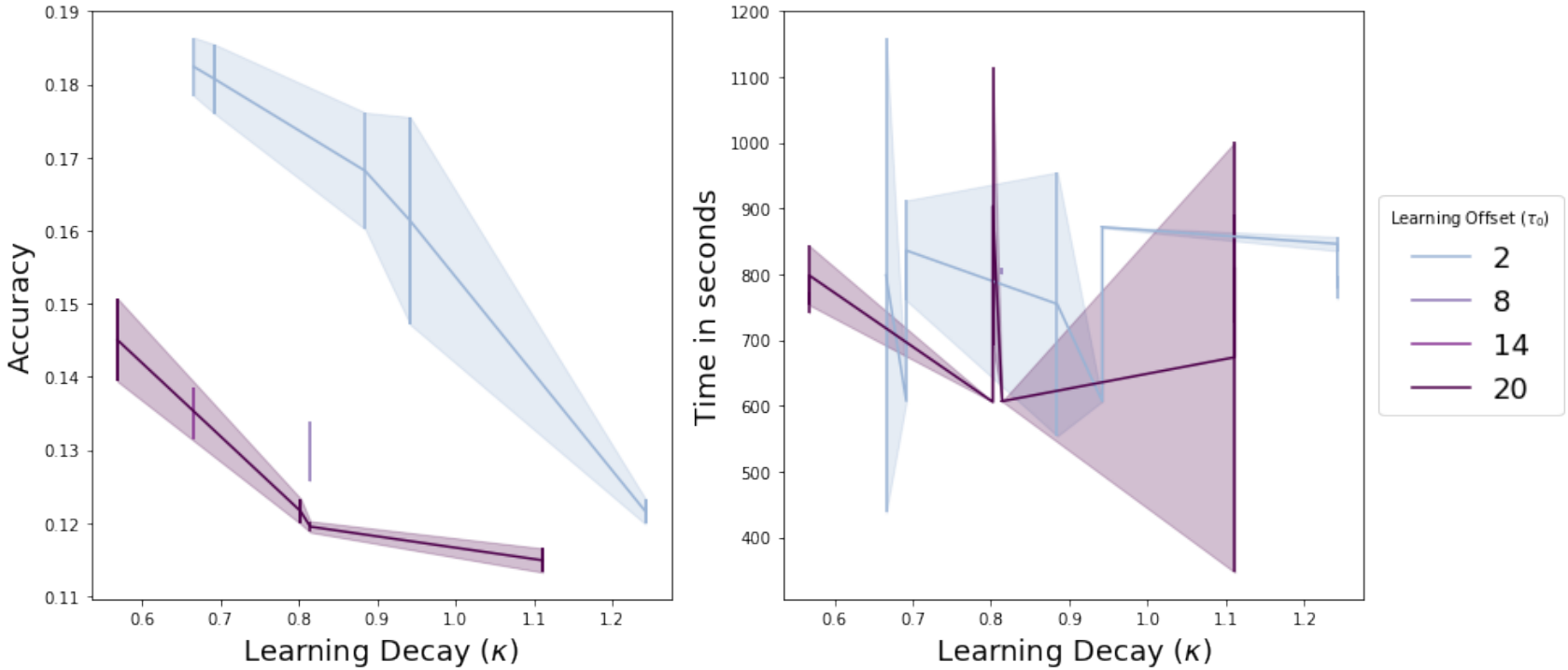


Figure 10: Accuracy and time are plotted against the decay rate (κ) for various learning offsets (τ_0). Shaded regions represent one standard deviation in cross-validation.

doc2vec

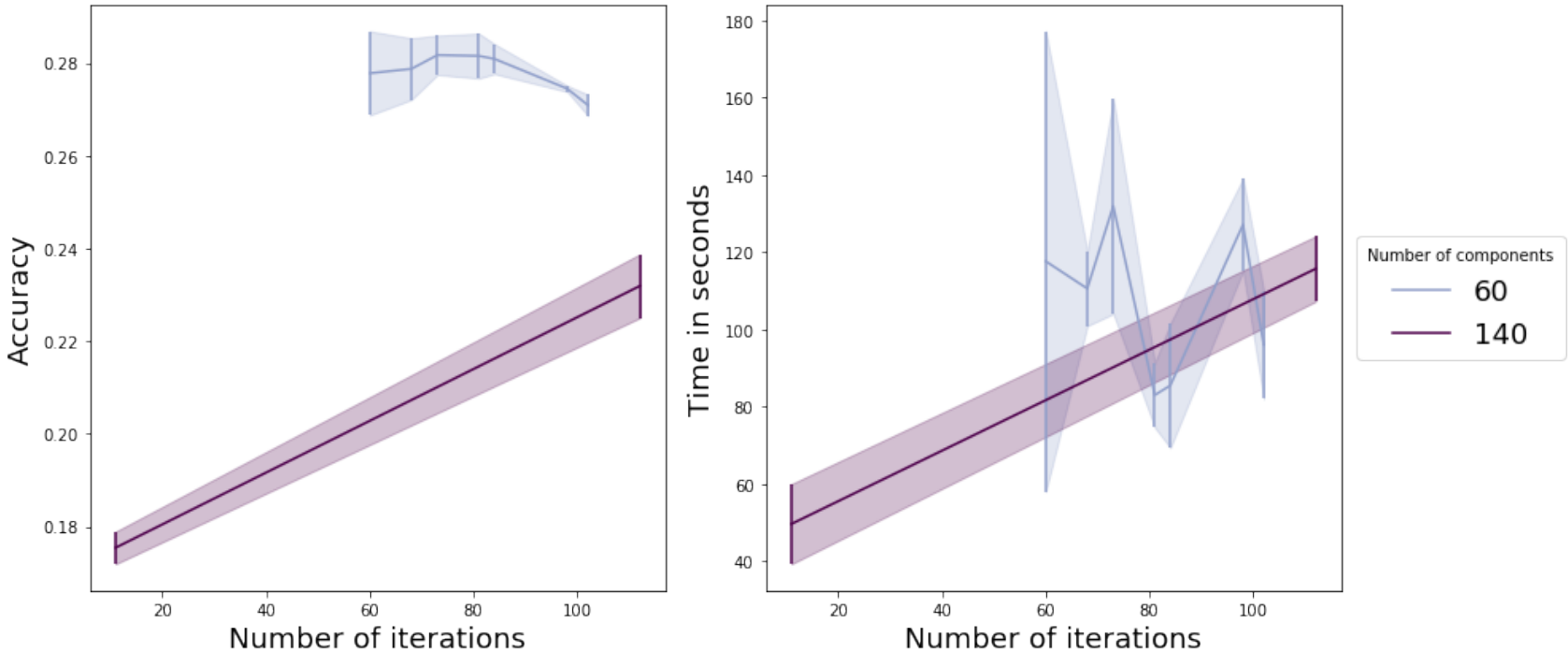


Figure 11: Accuracy and time are plotted against the number of training iterations for various numbers of components. Shaded regions represent one standard deviation in cross-validation.

Results of Hyperparameter Tuning

- KNN: 5 neighbors, ''distance' updating
- LSA: 60 topics
- LDA: Learning rate 0.51, Learning offset 2
- doc2vec: 60 topics, 75 epochs

Clustering: K-Means

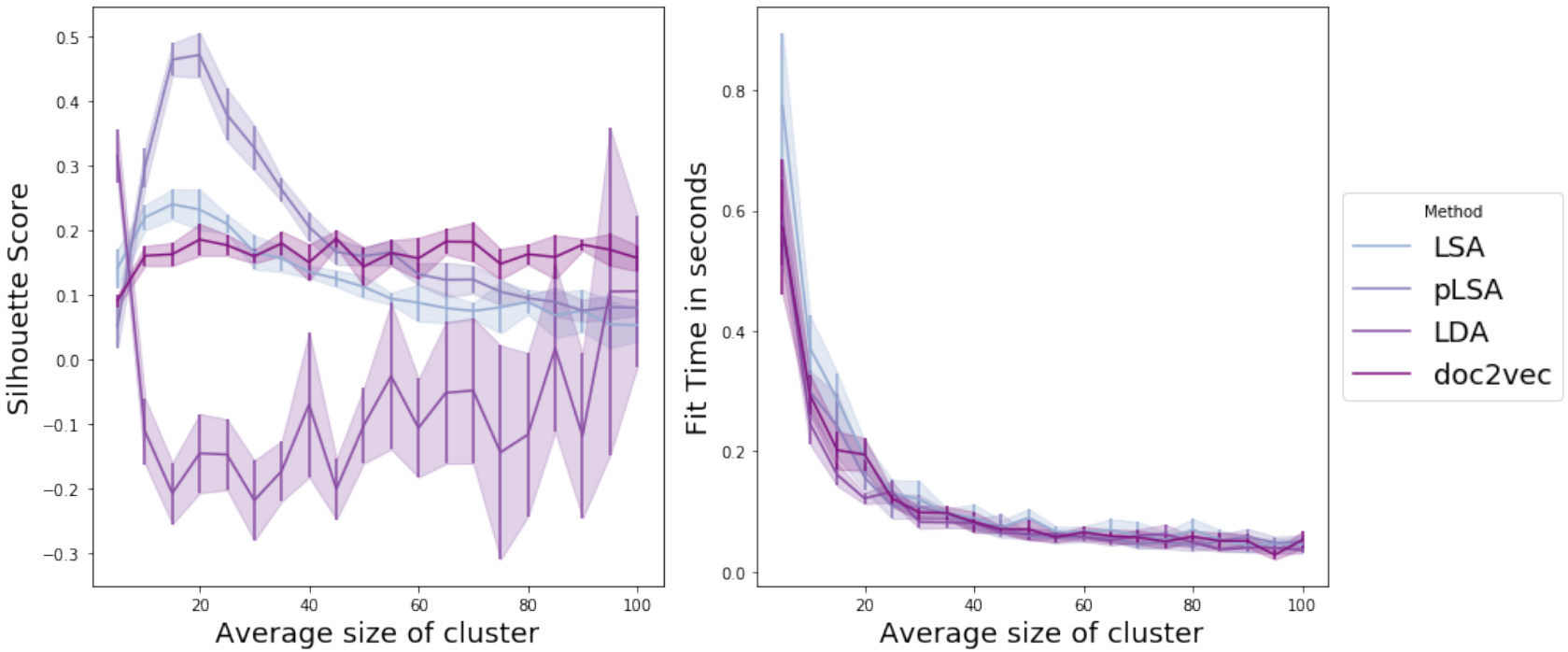


Figure 12: Silhouette score and time are plotted against the average size of each cluster for various feature extraction methods, using online K-Means. Shaded regions represent one standard deviation in cross validation.

Clustering: Birch

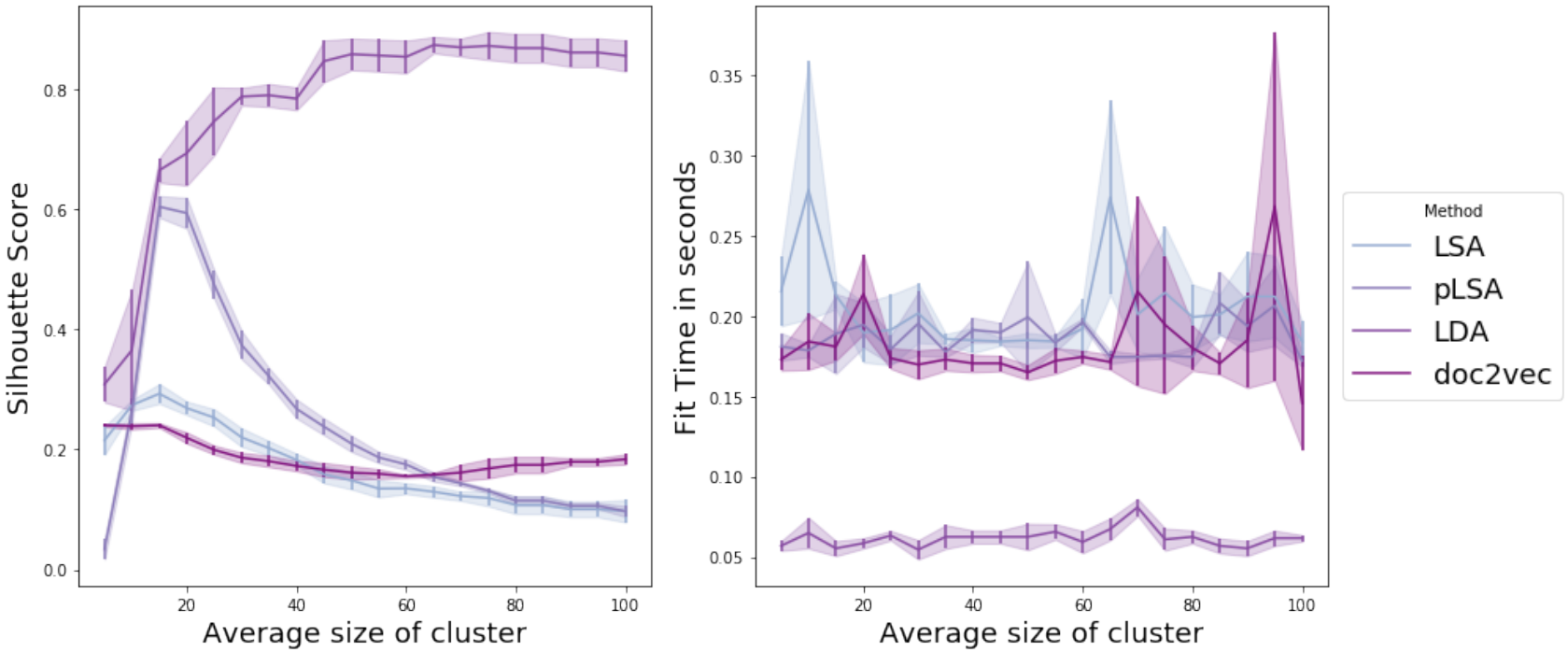


Figure 13: Silhouette score and time are plotted against the average size of each cluster for various feature extraction methods, using Birch. Shaded regions represent one standard deviation in cross validation.

Internal Evaluation of Clustering

Table 1: Silhouette Scores from various Methods

	K-Means	Birch
LSA	0.246	0.265
pLSA	0.359	0.420
LDA	0.321	0.609
doc2vec	0.114	0.133

Example Cluster: Cryptocurrency Subreddits

LSA: { 'vertcoin', 'ethereum', 'CryptoCurrency', 'Bitcoin', 'ethtrader', 'Monero', 'waltonchain', 'BitcoinMarkets', 'Iota', 'BitcoinAll', 'Ripple', 'NEO', 'btc' }

doc2vec: { 'vertcoin', 'ethereum', 'CryptoCurrency', 'Bitcoin', 'ethtrader', 'Monero', 'waltonchain', 'BitcoinMarkets', 'Iota', 'BitcoinAll', 'Ripple', 'NEO', 'btc' }

Figure 14: Most common words in cryptocurrency cluster (doc2vec, Birch)

Example Cluster: Music Listener Subreddits

{ 'Kanye', 'TaylorSwift', 'FrankOcean', 'indieheads', 'hiphopheads', 'bangtan', 'Metalcore', 'listentothis', 'Eminem', 'Metal', 'popheads', 'radiohead', 'deathgrips', 'kpop', 'Music' }

Figure 15: Most common words in music listener cluster (doc2vec, Birch)

Example Cluster: Music Maker Subreddits

{ 'synthesizers', 'makinghiphop', 'WeAreTheMusicMakers', 'brandnew', 'edmproduction', 'Guitar', 'vinyl', 'guitarpedals', 'headphones' }

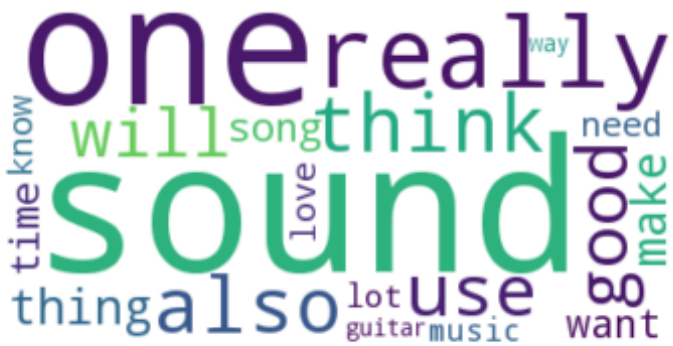


Figure 16: Most common words in music maker cluster (doc2vec, Birch)

Example Recommendation: Cities

Given /r/Fitness, I recommend...
/r/xxfitness, at a distance of 0.21
/r/bodybuilding, at a distance of 0.22
/r/weightroom, at a distance of 0.36
/r/orangetheory, at a distance of 0.39
/r/powerlifting, at a distance of 0.47

Figure 17: Nearest neighbors search for /r/Fitness (doc2vec, Annoy)

Conclusions

- Ultra-cheap LSA is competitive with complex architectures
- Subreddit discovery based on common topics of discussion
- Ordinary users can find new communities
  - Begin with known pages, like /r/bitcoin
  - Find a wealth of other pages
- Site admins can find and follow problematic communities
  - No user data: duplicate/fake accounts do not affect results
  - Growing concerns of bullying, exploitation, and propaganda
- Same principles apply to other social media platforms

Future Work

- More hyperparameter tuning
- Transfer learning (under active development)
- 50x more data every month (5000x in model tuning)
  - Re-work algos for distributed data
  - Execute with, e.g., Spark
  - Update with streaming data
- Alternate methods: FastText, Faiss, NMSLib
- Market basket analysis: sets of users' frequent subreddits
- Recommend *posts* as well as subreddits
- User interface

References

- Bird, Steven, Edward Loper and Ewan Klein (2009), *Natural Language Processing with Python*. OReilly Media Inc.
- Blei, David et al. "Latent Dirichlet Allocation". In: *Journal of Machine Learning Research* 3 (2003), pp. 993–1022. `https://arXiv.org/abs/1111.6189v1`
- Le, Quoc and Tomas Mikolov. Distributed Representations of Sentences and Documents. `https://arxiv.org/abs/1405.4053`
- Řehůřek, Radim and Petr Sojka (2010). "Software Framework for Topic Modelling with Large Corpora". In: *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pp. 45–50. ELRA: Valletta, Malta. `http://is.muni.cz/publication/884893/en`
- Scikit-learn: Machine Learning in Python, Pedregosa et al., *JMLR* 12, pp. 2825-2830, 2011.
- Spotify (2018). Annoy. Open source. `github.com/spotify/annoy`