Finding Similar Repositories in Github

Jingci Wang
MS in Data Science
Khoury College of Computer Sciences

Motivation

Finding relevant projects is beneficial for developers in case of

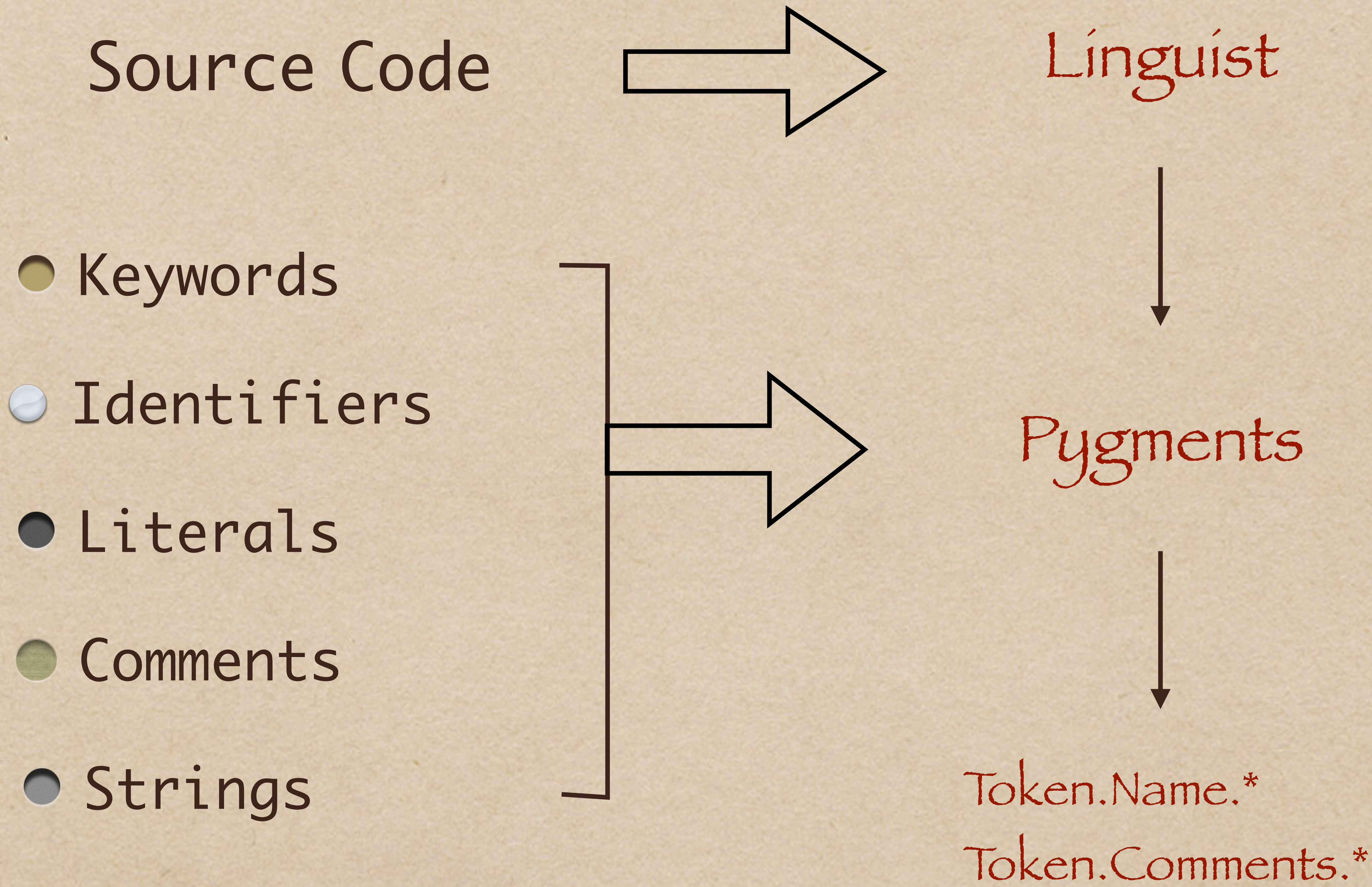
- Reuse existing functions
- Explore ideas of possible features
- Analyze the requirements and possible implementations for their own projects

Objective

- Given a single software repository, find repositories that are most similar to it

Possible Solutions

- Dig patterns in GitHub users's history behaviors
- Find similarity in Readme Files
- Find similarity in Source Code



Source Code
Linguist
Keywords
Identifiers
Literals
Comments
Strings
Pygments
Token.Name.*
Token.Comments.*

Data Preparation

- Treat every file as a sentence, repo as a document
- Scraped source code identifiers and comments from GitHub repositories
- Split identifiers foo_BAR —> (foo, bar)
- Stemmed identifiers —> Bag-of-Words
- Removed repos with less 50 different identifiers or more than 1000000 words in total
- Resulted in 1785 repos

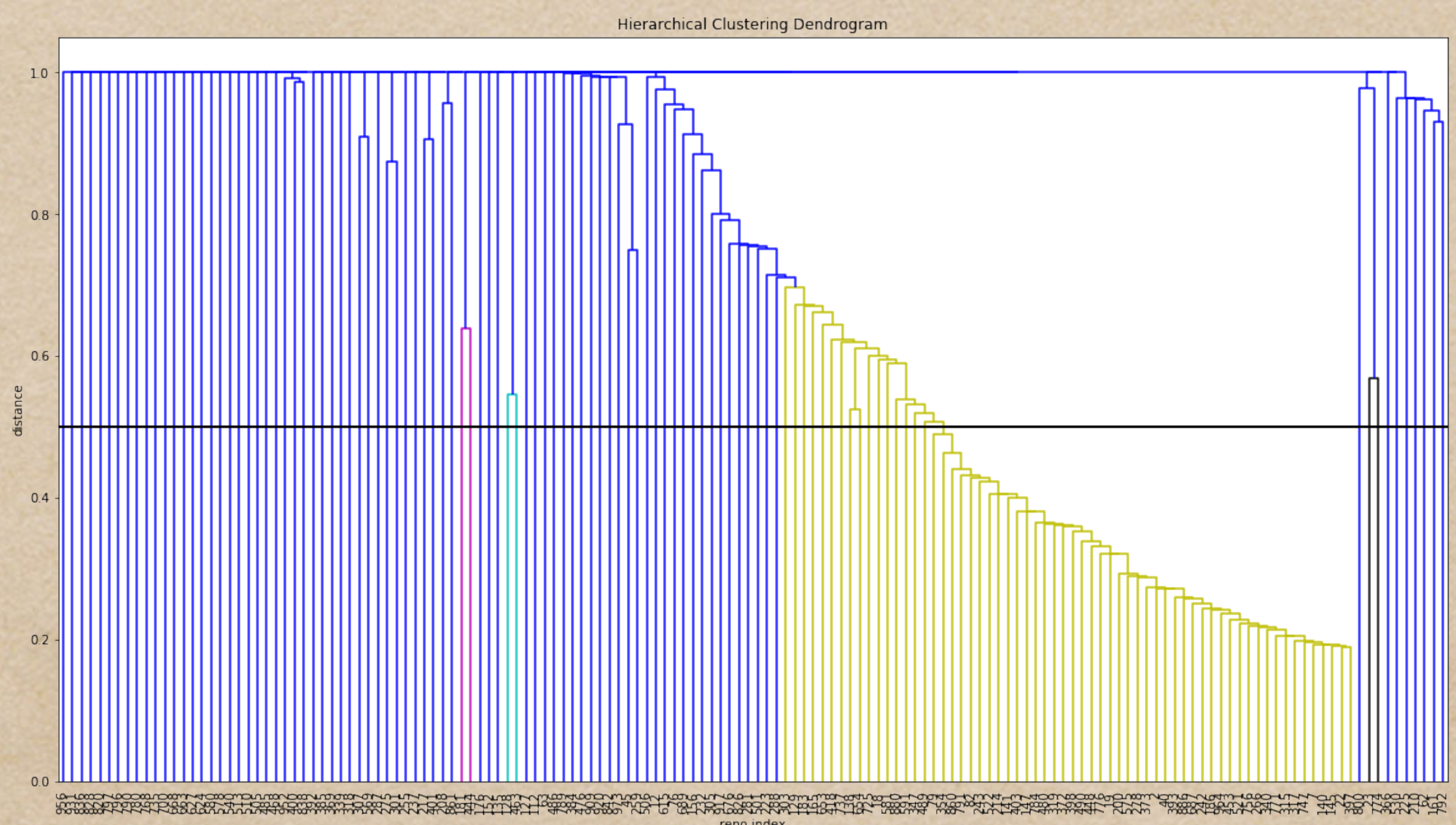
Evaluation Set

__956 Pairs

Thanks to DéjàVu

cloneId	#clonedFiles	#totalFiles	clonePercent	hostId	#affectedFiles	#totalFiles	affectPercent
7	4	18	22.22	521	266	434	61.26

$$sim(repo_1, repo_2) = \frac{\#repo_1Files \times cloned\% + \#repo_2Files \times affected\,\%}{\#repo_1Files + \#repo_2Files}$$



Vectorization

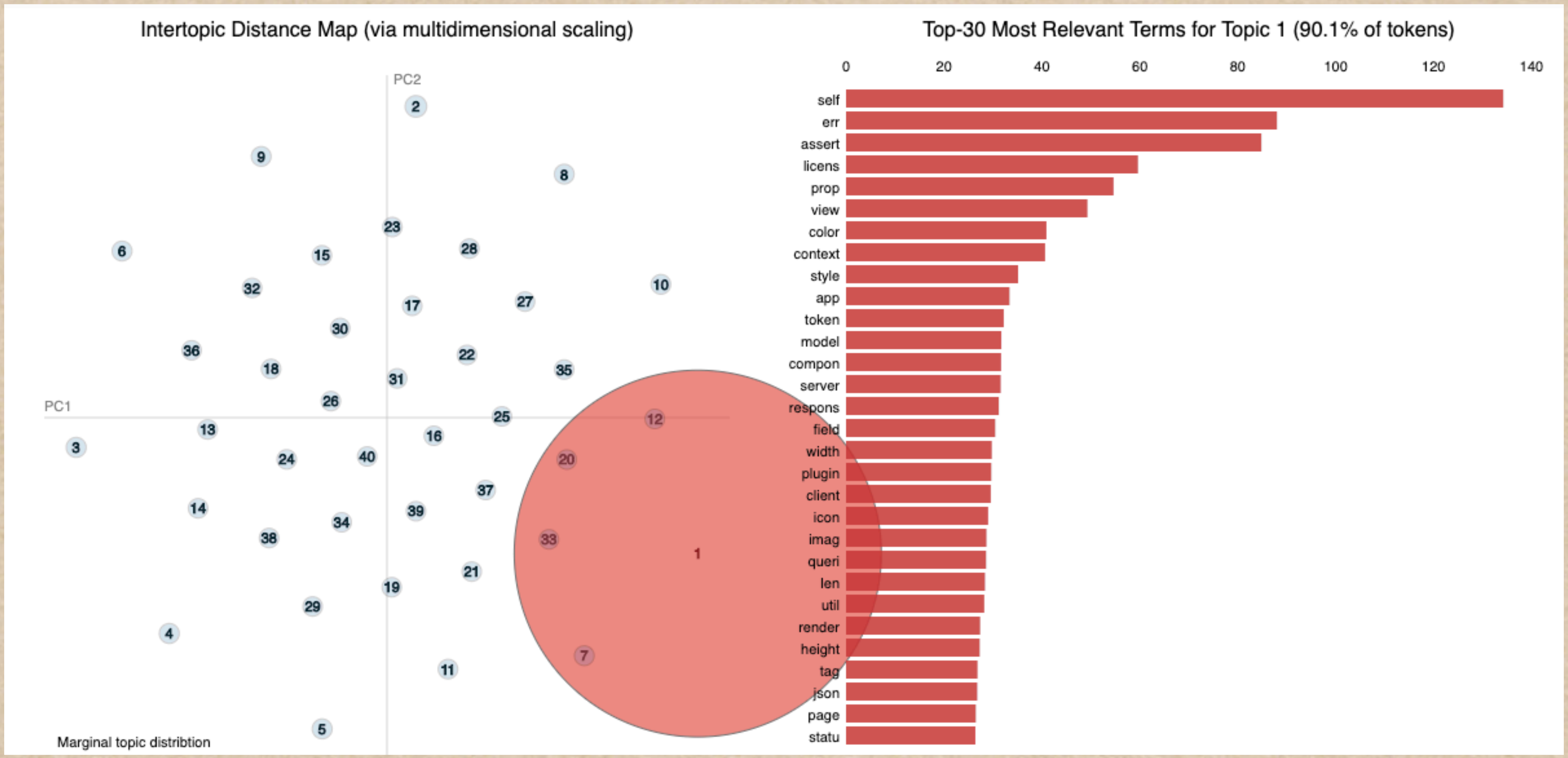
- CountVectorizer __more than 150000 cols, 99.6% sparsity

```
aa
aaa
aaaa
aaaaa
aaaaaa
aaaaaaaaaaaaa
aaaaaaaaaaaaaa
aaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aa
aaaaaaaaaabbbbbbbbbb
aaaaae
```

- Only keep words which appear in more than 5 repos but less 75% of repos

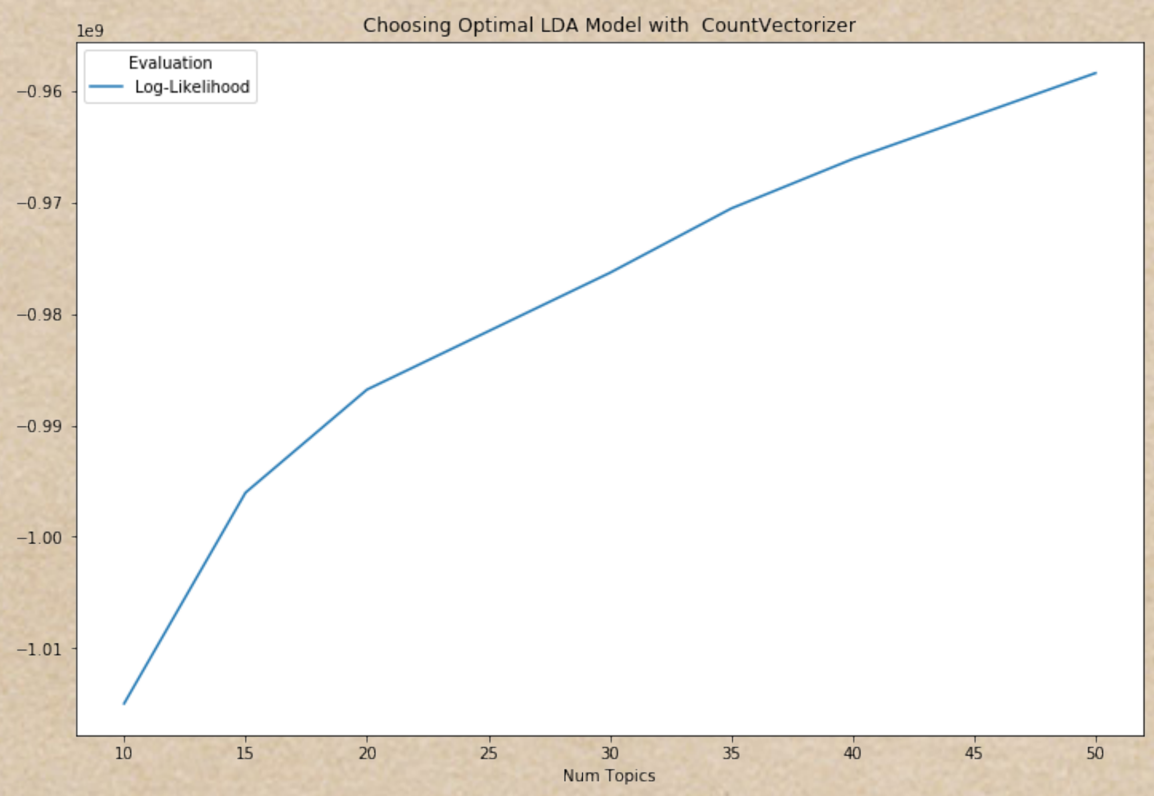
__45600, 98% sparsity

Problem with TFIDF

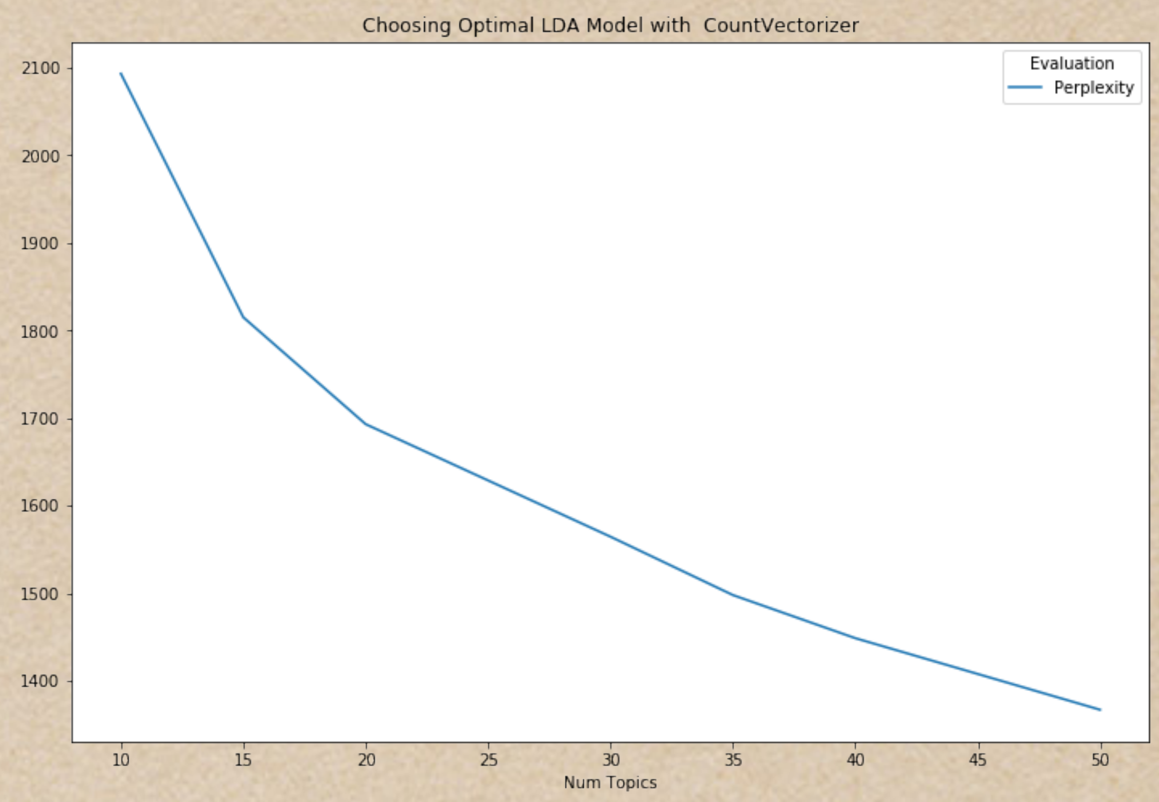


Dominant topic in each repository is the same

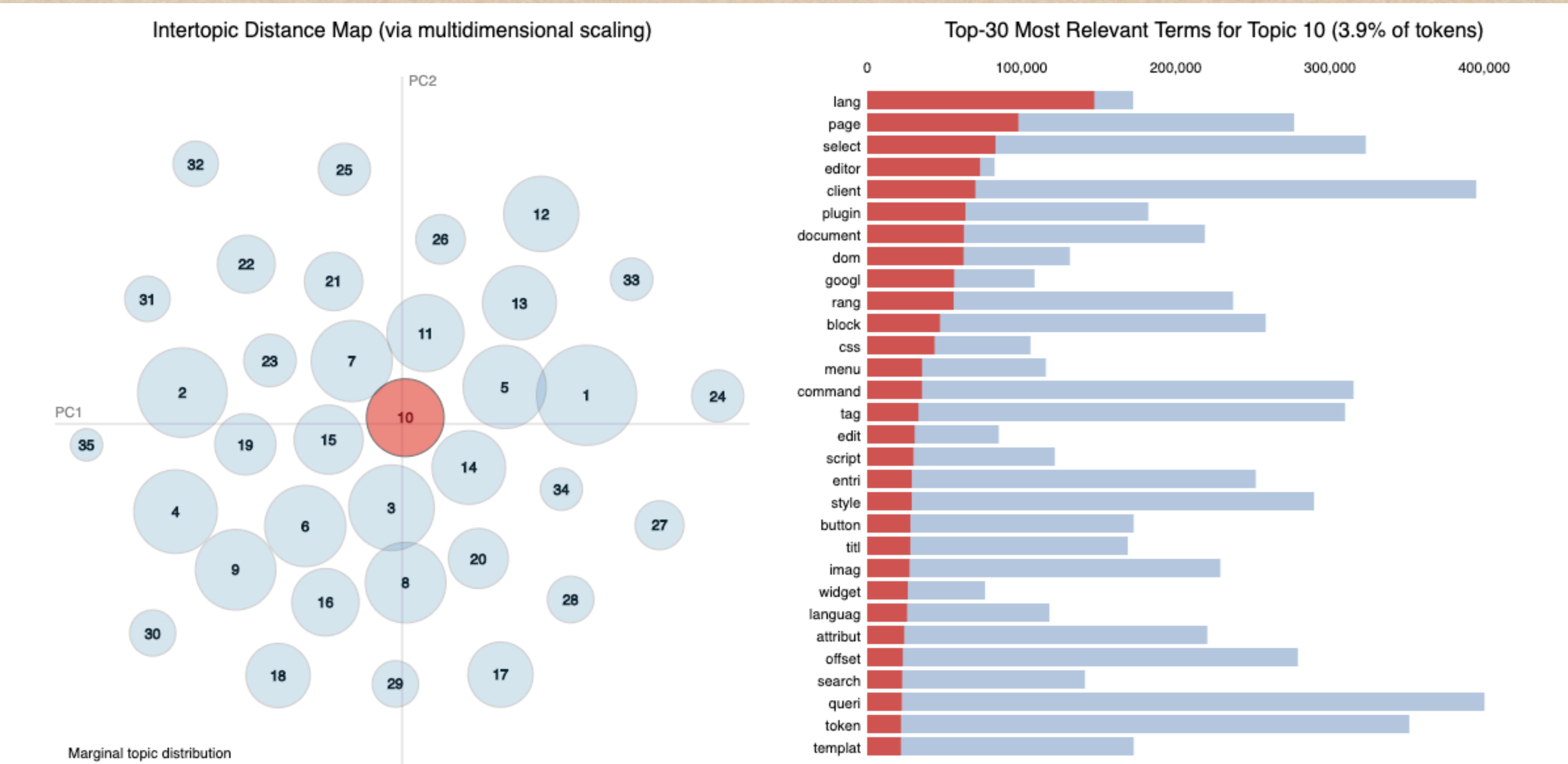
Topic Modeling



Log-likehood v.s. #Topics

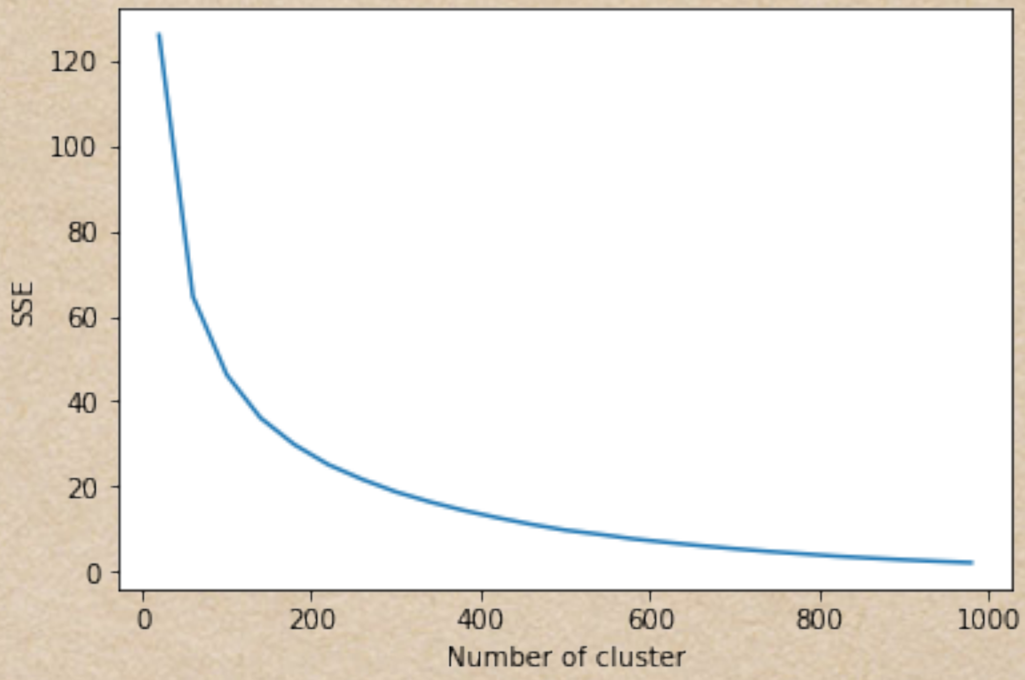


Perplexity v.s. #Topics



Perplexity	Log-Likelihood
911.8168	-678301280.3942

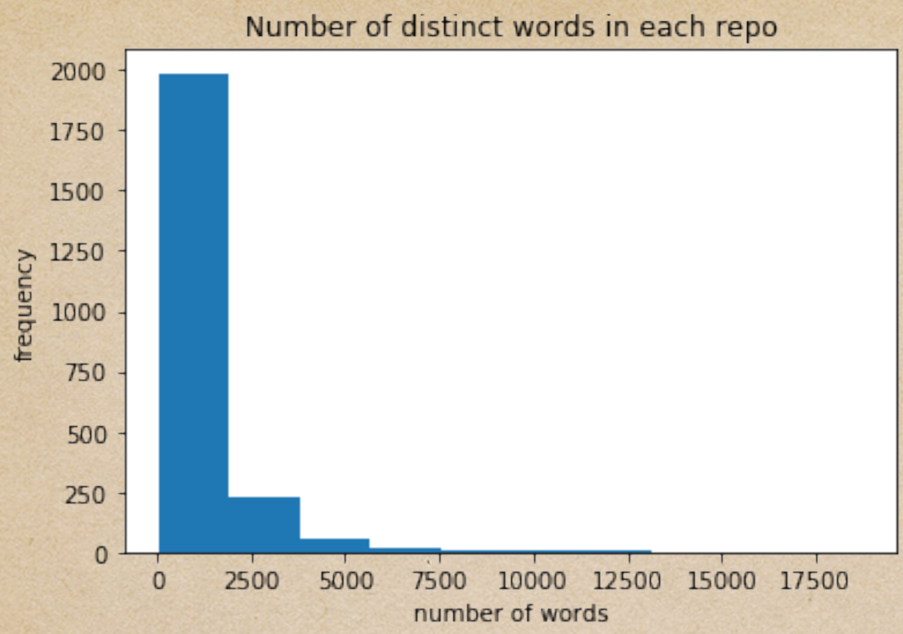
Perform KMeans resulting in 150 clusters

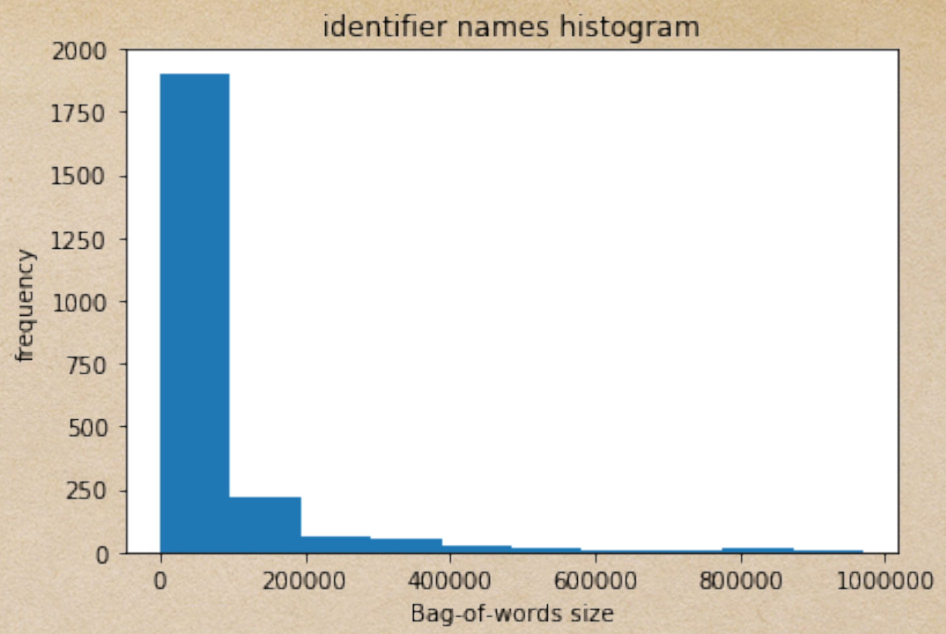


$$Purity_i = \frac{max_j N_j}{C_i}$$

Recall	Precision	Homogeneity	Completeness	SSE
0.07	0.56	0.880	0.738	74.3105

- Running time suffers for a giant dataset

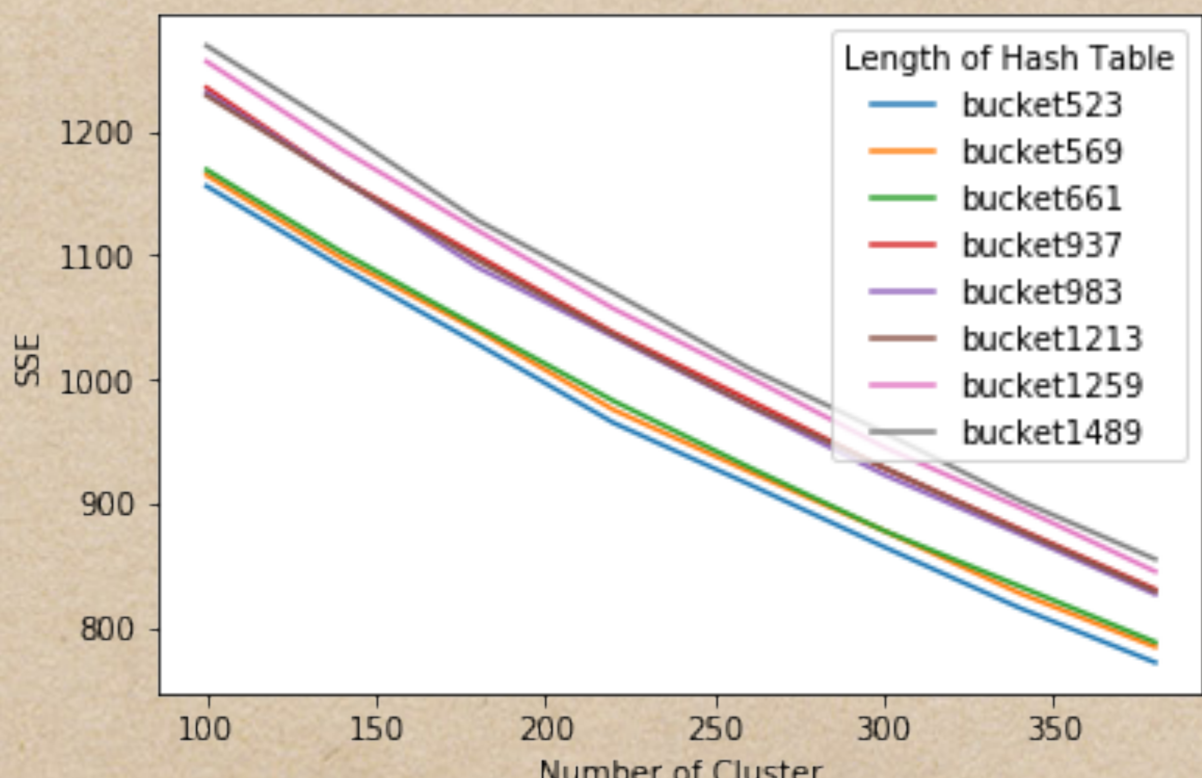




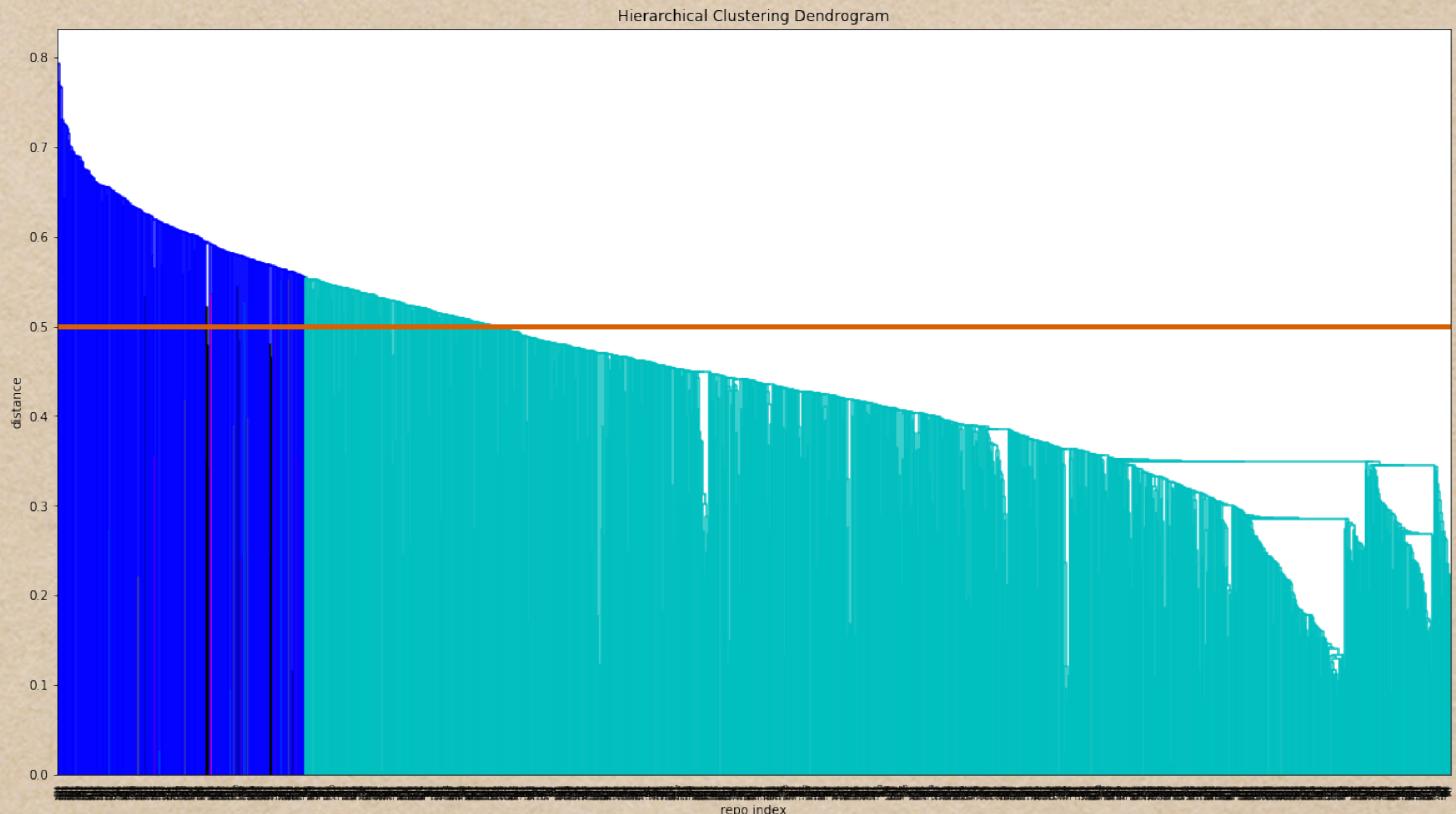
	#Different Words	#Words
Mean	1217	74356
25%	301	4124
50%	654	17201
75%	1400	64258

Hash Method

Identifier



1785 by 523 Matrix



Hierarchical Clustering Dendrogram
distance
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
repo index

	Recall	Precision	Homogeneity	Completeness	SSE
Hash	0.44	0.51	0.526	0.767	74.3105
Hash (norm)	0.15	0.47	0.789	0.723	3E+10
LDA	0.07	0.56	0.880	0.738	836.38

Results with only identifier names

Homogeneity	Completeness
0.6099	0.5748

LDA v.s. Hash

	Recall	Precision	Homogeneity	Completeness
Hash	0.44	0.45	0.510	0.754
Hash (norm)	0.10	0.51	0.810	0.724
LDA	0.07	0.54	0.869	0.731

Results adding comments

Examples

WordPress/WordPress
Jumilla/wordpress-plus
dxw/wordpress
mhoofman/wordpress-heroku
owen2345/camaleon-cms

torch/nn
pytorch/pytorch
jcjohnson/neuralstyle
keras-team/keras
tensorflow/models

twitter/mysql
Tokutek/mysql-5.5
Tokutek/mariadb-5.5
facebook/mysql-5.6
therecluse26/PHP-Login

Thanks