ML HW4

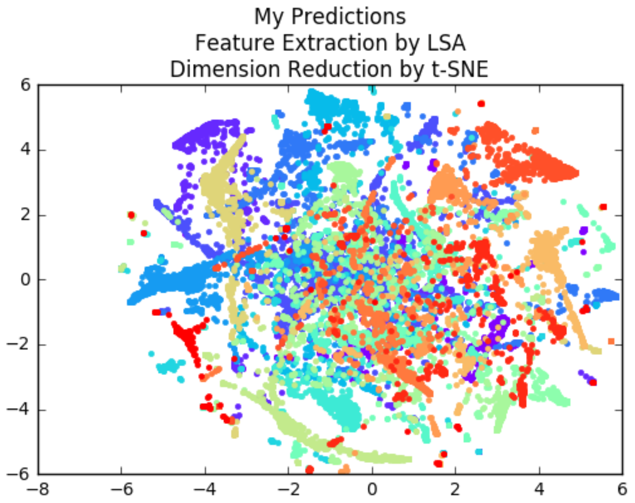
R04942070 吳冠融

1. Analyze the most common words in the clusters.

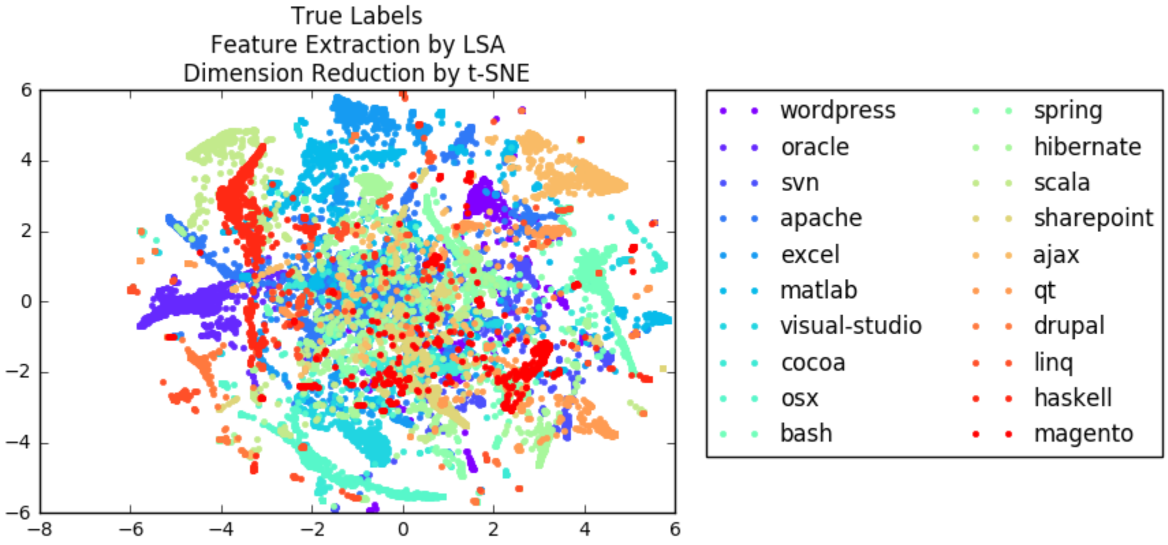
label	name	most common 10 words
1	wordpress	wordpress, posts, wp, category, post, blog, page, theme, plugin, php
2	oracle	oracle, sql, pl, ora, table, procedure, query, database, using, stored
3	svn	svn, subversion, repository, commit, branch, revision, files, file, externals, repositories
4	apache	apache, mod, rewrite, htaccess, redirect, php, server, rewriterule, url, using
5	excel	excel, vba, cell, macro, workbook, cells, worksheet, sheet, file, data
6	matlab	matlab, matrix, plot, function, using, array, vector, vectors, image, file
7	visual-studio	studio, visual, vs, project, solution, build, projects, files, file, code
8	cocoa	cocoa, objective, nstableview, nsstring, nsview, core, window, nsoutlineview, app, xcode
9	osx	mac, os, osx, cocoa, terminal, leopard, application, xcode, app, file
10	bash	bash, script, file, shell, command, files, line, variable, using, sed
11	spring	spring, bean, hibernate, beans, mvc, using, security, annotations, annotation, aop
12	hibernate	hibernate, hql, mapping, criteria, jpa, query, using, annotations, entity, join
13	scala	scala, actors, java, lift, type, actor, using, class, list, trait
14	sharepoint	sharepoint, moss, web, site, list, webpart, wss, workflow, custom, document
15	ajax	ajax, jquery, javascript, asp, using, php, request, net, page, response
16	qt	qt, creator, widget, qwidget, qmake, signals, application, using, window, qtableview
17	drupal	drupal, node, cck, module, views, form, taxonomy, theme, page, content
18	linq	linq, sql, query, using, iqueryable, ienumerable, xml, join, select, group
19	haskell	haskell, type, function, ghc, list, cabal, monad, functional, ghci, parsec
20	magento	magento, product, products, category, price, checkout, admin, cart, customer, custom

2. Visualize the data

Using LSA for featue extraction and t-SNE for dimension reduction.
My cluster predictions (private score = 0.75)



True label



We can see that the clustering method works well. In most cases, my prediction is the same as the true labels.

3. Compare different feature extraction methods.

Method	Config	Private Score

Bag of Words	remove stopwords
	max_features = 5000
	cluster = 20
		0.15187

Bag of Words + SVG	remove stopwords
	max_features = 5000
	n_components = 20
	cluster = 80
		0.77791

TF-IDF	remove stopwords cluster = 20	0.22261

LSA
(TF-IDF + SVG)
	remove stopwords
	n_components = 20
	cluster = 80
		0.83483

Word2vec	trained with docs and titles
	dimension = 500
	cluster = 20
		0.45196

Word2vec + SVG	trained with docs and titles
	dimension = 500
	n_components = 20
	cluster = 20
		0.42403

Conclusions

- LSA works best in all my experiments.
- TF-IDF is better than BoW, but not much better. I removed the stopwords before applying BoW of TF-IDF, maybe that’s why TF-IDF didn’t outperform BoW much.
- Using SVG to reduce dimensions before clustering improves BoW and TF-IDF a lot, but somehow doesn’t improve Word2vec.
- For word2vec, I used the average of word vectors in a title as the feature of the title. There may be better way to generate the feature of a sentence so that word2vec can work better.

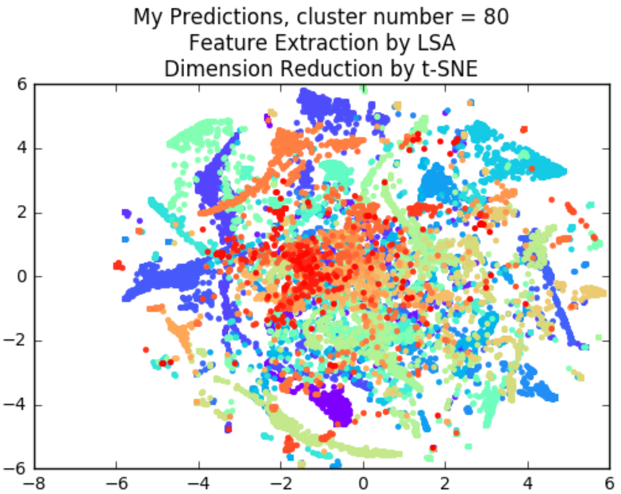
4. Try different cluster numbers and compare them.

Using LSA, n_components=20

cluster=	20	50	80	100	120	150	200
private score	0.60416	0.80363	0.83483	0.82849	0.82554	0.81697	0.79280

Surprisingly, the best cluster number is about 80, which is far from the number of labels.

I visualize the clusters. It seems that the boundaries between clusters are much clear. Maybe that’s the reaason why it clusters the data better.


My Predictions, cluster number = 80
Feature Extraction by LSA
Dimension Reduction by t-SNE