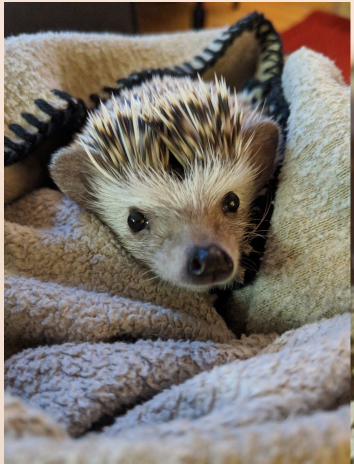
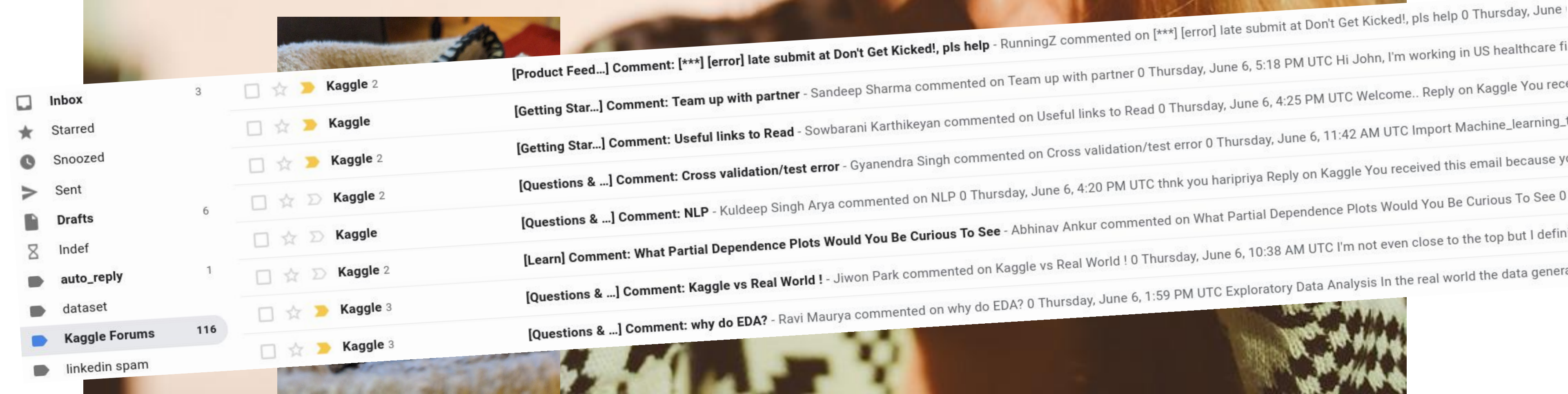
Unsupervised NLP Techniques & The Kaggle Forums

Rachael Tatman, Kaggle





Kaggle 2 **[Product Feed...] Comment: [***] [error] late submit at Don't Get Kicked!, pls help** - RunningZ commented on [***] [error] late submit at Don't Get Kicked!, pls help 0 Thursday, June

Kaggle **[Getting Star...] Comment: Team up with partner** - Sandeep Sharma commented on Team up with partner 0 Thursday, June 6, 5:18 PM UTC Hi John, I'm working in US healthcare fi

Kaggle 2 **[Getting Star...] Comment: Useful links to Read** - Sowbarani Karthikeyan commented on Useful links to Read 0 Thursday, June 6, 4:25 PM UTC Welcome.. Reply on Kaggle You rece

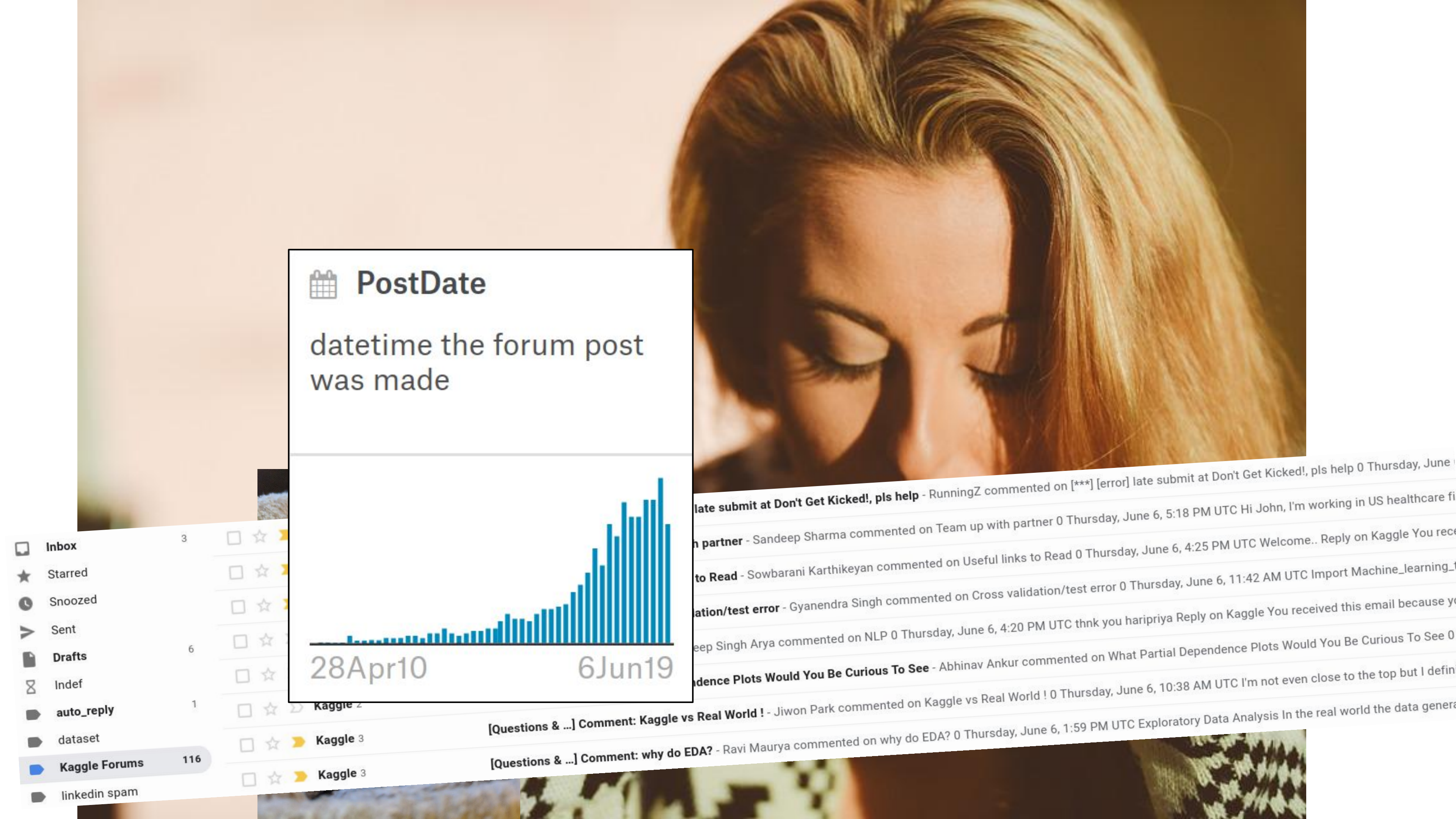
Kaggle 2 **[Questions & ...] Comment: Cross validation/test error** - Gyanendra Singh commented on Cross validation/test error 0 Thursday, June 6, 11:42 AM UTC Import Machine_learning_

Kaggle **[Questions & ...] Comment: NLP** - Kuldeep Singh Arya commented on NLP 0 Thursday, June 6, 4:20 PM UTC thnk you haripriya Reply on Kaggle You received this email because y

Kaggle 2 **[Learn] Comment: What Partial Dependence Plots Would You Be Curious To See** - Abhinav Ankur commented on What Partial Dependence Plots Would You Be Curious To See 0

Kaggle 3 **[Questions & ...] Comment: Kaggle vs Real World !** - Jiwon Park commented on Kaggle vs Real World ! 0 Thursday, June 6, 10:38 AM UTC I'm not even close to the top but I defin

Kaggle 3 **[Questions & ...] Comment: why do EDA?** - Ravi Maurya commented on why do EDA? 0 Thursday, June 6, 1:59 PM UTC Exploratory Data Analysis In the real world the data genera



PostDate

datetime the forum post was made

late submit at Don't Get Kicked!, pls help - RunningZ commented on [***] [error] late submit at Don't Get Kicked!, pls help 0 Thursday, June

h partner - Sandeep Sharma commented on Team up with partner 0 Thursday, June 6, 5:18 PM UTC Hi John, I'm working in US healthcare fi

to Read - Sowbarani Karthikeyan commented on Useful links to Read 0 Thursday, June 6, 4:25 PM UTC Welcome.. Reply on Kaggle You rece

ation/test error - Gyanendra Singh commented on Cross validation/test error 0 Thursday, June 6, 11:42 AM UTC Import Machine_learning_

eep Singh Arya commented on NLP 0 Thursday, June 6, 4:20 PM UTC thnk you haripriya Reply on Kaggle You received this email because y

dence Plots Would You Be Curious To See - Abhinav Ankur commented on What Partial Dependence Plots Would You Be Curious To See 0

Kaggle 2

Kaggle 3 [Questions & ...] Comment: Kaggle vs Real World ! - Jiwon Park commented on Kaggle vs Real World ! 0 Thursday, June 6, 10:38 AM UTC I'm not even close to the top but I defin

Kaggle 3 [Questions & ...] Comment: why do EDA? - Ravi Maurya commented on why do EDA? 0 Thursday, June 6, 1:59 PM UTC Exploratory Data Analysis In the real world the data genera

Problem: I can't keep reading all the forum posts on Kaggle with my human eyeballs

Problem: I can't keep reading all the forum posts on Kaggle with my human eyeballs

Solution: Unsupervised clustering to summarize common topics & user concerns

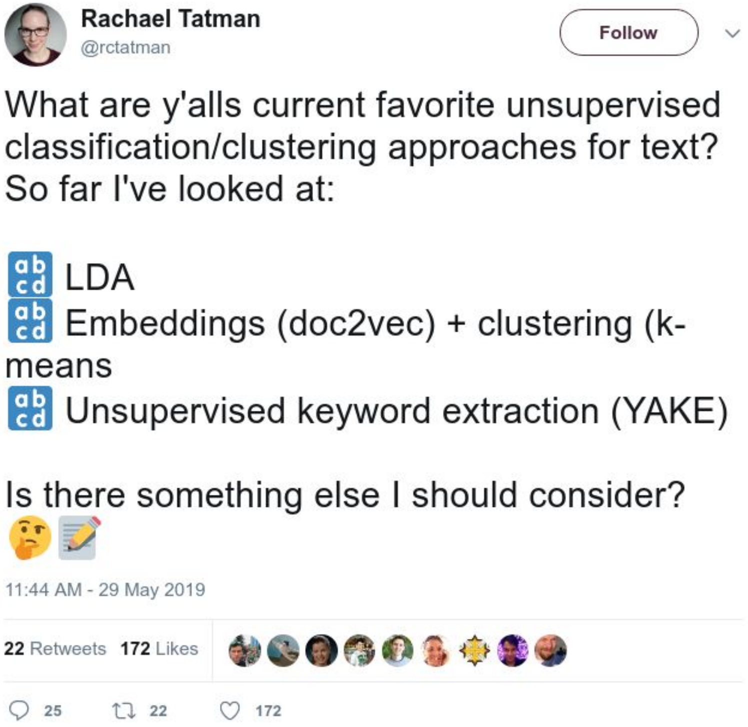
Problem: I can't keep reading all the forum posts on Kaggle with my human eyeballs

Solution: Unsupe ring to summarize comm user concerns

Some ground rules:

- Needs to be in Python or R
  - I’m livecoding the project in Kernels & those are the only two languages we support
  - I just don’t want to use Java or C++ or Matlab whatever
- Needs to be fast to retrain or add new classes
  - New topics emerge very quickly (specific bugs, competition shakeups, ML papers)
  - I'll probably have to re-run it daily or weekly
  - Eventually... streaming?
- Want to avoid large/weird dependencies
  - “Oh, that’s just some .jar I downloaded from a random website. The code doesn’t run without it and I’m sure it’s fine to just stick in our codebase.”
- Clusters/topics should be easily interpretable

I asked on Twitter!



Rachael Tatman
@rctatman

What are y'alls current favorite unsupervised classification/clustering approaches for text? So far I've looked at:

🔡 LDA
🔡 Embeddings (doc2vec) + clustering (k-means
🔡 Unsupervised keyword extraction (YAKE)

Is there something else I should consider? 🤔📝

11:44 AM - 29 May 2019

22 Retweets 172 Likes

Lots of good ideas!

Three main bins:

- End-to-end solutions
- Suggestions for feature engineering + clustering
- Misc. tips & tricks (ex: embeddings -> PCA -> remove 1st principle component)

End-to-end solutions

- Gensim
  - ✓ In Python, no weird dependencies
  - ✓ Old standby that incorporates a looot of differents methods
  - ✓ Don’t need whole corpus in memory (but mine’s not that big)
  - ❌ Under LGPL (probably fine for prototyping, but might need to set up meetings with legal if I’m using it for work stuff & that’s more overhead than I want)
- BigARTM
  - ✓ Can incorporate multiple objectives at once (sparsing, smoothing, decorrelation, etc.)
  - ❌ Weird dependency/install process (it’s a C++ library with a Python API)
- TopSBM
  - ✓ Came highly recommended: “Scary good”
  - ❌ Weird dependency (graph-tool, which is C++ with a Python wrapper)

Feature Engineering: Words to numbers

- Traditional Topic Modelling Approaches
  - **LDA**: Slow, hard to interpret, not my fave
  - **pLSA**: Cheaper version of LSA, tends to overfit
  - **tf-idf**: Hard to interpret, my texts (forums posts) are too short
- Embeddings
  - **GloVe**: considers context, can’t handle new words
  - **Word2vec**: doesn’t handle small corpuses very well, very fast to train
  - **fasttext**: can handle out of vocabulary words (extension of word2vec)
- Contextual embeddings (don’t think I have enough data to train my own…)
  - **ELMO, BERT, etc.**: I consider these more of a replacement for language models
  - **USE embeddings**: Not super familiar with this but looks useful for applying to sentence similarity

Feature Engineering: Words to numbers

- Traditional Topic Modelling Approaches
  - **LDA**: Slow, hard to interpret, not my fave
  - **pLSA**: Cheaper version of LSA, tends to overfit
  - **tf-idf**: Hard to interpret, my texts (forums posts) are too short
- Embeddings
  - **GloVe**: considers context, can't handle new words
  - **Word2vec**: doesn't handle small corpuses very well, very fast to train
  - **fasttext**: can handle out of vocabulary words (extension of word2vec)
- Contextual embeddings (don't think I have enough data to train my own…)
  - **ELMO, BERT, etc.**: I consider these more of a replacement for language models
  - **USE embeddings**: Not super familiar with this but looks useful for applying to sentence similarity

Feature Engineering: Dimensionality Reduction

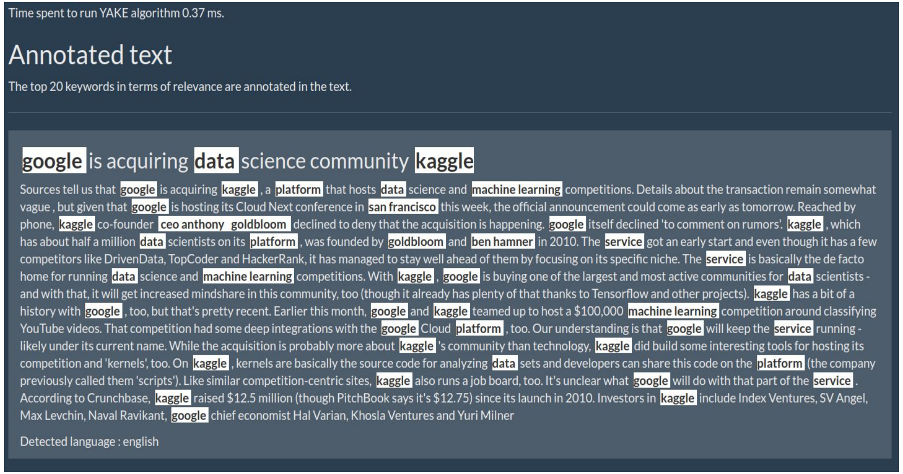
- UMAP:
  - Recommended to me by, among other people, Leland McInnes, the researcher who developed it 😂 (he suggested using hellinger distance)
  - Similar to t-SNE but can also be used for non-linear dimension reduction
  - Something about manifolds? (The math's a little over my head, tbh)
- PCA
  - OG dimensionality reduction (paper is from 1901!) but on its own maybe not the best
  - Trick: remove first principal component as a way to reduce the weight of "expected" words
    - (from Arora (2018) 'A simple but tough to beat baseline for sentence embeddings')

Feature Engineering: Dimensionality Reduction

- UMAP:
  - Recommended to me by, among other people, Leland McInnes, the researcher who developed it 😂 (he suggested using hellinger distance)
  - Similar to t-SNE but can also be used for non-linear dimension reduction
  - Something about manifolds? (The math's a little over my head, tbh)
- PCA:
  - OG dimensionality reduction (paper is from 1901!) but on its own maybe not the best
  - Trick: remove first principal component as a way to reduce the weight of "expected" words
    - (from Arora (2018) 'A simple but tough to beat baseline for sentence embeddings')

Wildcard!

- Unsupervised keyword extraction: YAKE
    - Extracts keywords from single texts
    - Could use it as dimensionality reduction
    - Keywords -> embeddings -> clustering?
    - One of their sample texts is about the Kaggle acquisition! 😊
    - Haven’t played around with it, but came highly recommended
    - `pip install git+https://github.com/LIAAD/yake`



Time spent to run YAKE algorithm 0.37 ms.

Annotated text

The top 20 keywords in terms of relevance are annotated in the text.

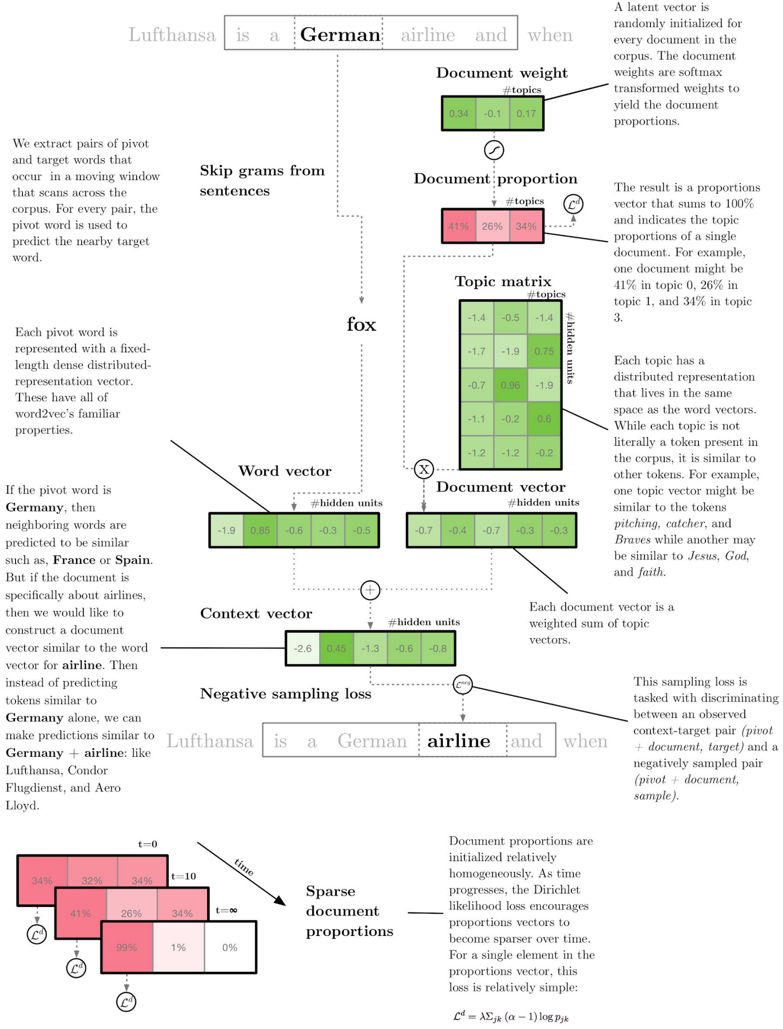
google is acquiring data science community kaggle

Sources tell us that google is acquiring kaggle , a platform that hosts data science and machine learning competitions. Details about the transaction remain somewhat vague , but given that google is hosting its Cloud Next conference in san francisco this week, the official announcement could come as early as tomorrow. Reached by phone, kaggle co-founder ceo anthony goldbloom declined to deny that the acquisition is happening. google itself declined 'to comment on rumors'. kaggle , which has about half a million data scientists on its platform , was founded by goldbloom and ben hamner in 2010. The service got an early start and even though it has a few competitors like DrivenData, TopCoder and HackerRank, it has managed to stay well ahead of them by focusing on its specific niche. The service is basically the de facto home for running data science and machine learning competitions. With kaggle , google is buying one of the largest and most active communities for data scientists - and with that, it will get increased mindshare in this community, too (though it already has plenty of that thanks to Tensorflow and other projects). kaggle has a bit of a history with google , too, but that's pretty recent. Earlier this month, google and kaggle teamed up to host a $100,000 machine learning competition around classifying YouTube videos. That competition had some deep integrations with the google Cloud platform , too. Our understanding is that google will keep the service running - likely under its current name. While the acquisition is probably more about kaggle 's community than technology, kaggle did build some interesting tools for hosting its competition and 'kernels', too. On kaggle , kernels are basically the source code for analyzing data sets and developers can share this code on the platform (the company previously called them 'scripts'). Like similar competition-centric sites, kaggle also runs a job board, too. It's unclear what google will do with that part of the service . According to Crunchbase, kaggle raised $12.5 million (though PitchBook says it's $12.75) since its launch in 2010. Investors in kaggle include Index Ventures, SV Angel, Max Levchin, Naval Ravikant, google chief economist Hal Varian, Khosla Ventures and Yuri Milner

Detected language : english

Wildcard!

- Lda2vec
  - Embeddings + topic models trained simultaneously
  - Developed at StitchFix 3ish years ago
  - Still pretty experimental but could be helpful
  - Under MIT license
  - Has a tutorial notebook
  - Might be very slow???



A latent vector is randomly initialized for every document in the corpus. The document weights are softmax transformed weights to yield the document proportions.

We extract pairs of pivot and target words that occur in a moving window that scans across the corpus. For every pair, the pivot word is used to predict the nearby target word.

The result is a proportions vector that sums to 100% and indicates the topic proportions of a single document. For example, one document might be 41% in topic 0, 26% in topic 1, and 34% in topic 3.

Each pivot word is represented with a fixed-length dense distributed-representation vector. These have all of word2vec's familiar properties.

Each topic has a distributed representation that lives in the same space as the word vectors. While each topic is not literally a token present in the corpus, it is similar to other tokens. For example, one topic vector might be similar to the tokens *pitching, catcher,* and *Braves* while another may be similar to *Jesus, God,* and *faith.*

If the pivot word is **Germany**, then neighboring words are predicted to be similar such as, **France** or **Spain**. But if the document is specifically about airlines, then we would like to construct a document vector similar to the word vector for **airline**. Then instead of predicting tokens similar to **Germany** alone, we can make predictions similar to **Germany** + **airline**: like Lufthansa, Condor Flugdienst, and Aero Lloyd.

Each document vector is a weighted sum of topic vectors.

This sampling loss is tasked with discriminating between an observed context-target pair *(pivot + document, target)* and a negatively sampled pair *(pivot + document, sample)*.

Document proportions are initialized relatively homogeneously. As time progresses, the Dirichlet likelihood loss encourages proportions vectors to become sparser over time. For a single element in the proportions vector, this loss is relatively simple:

$$\mathcal{L}^d = \lambda \Sigma_{jk} (\alpha - 1) \log p_{jk}$$

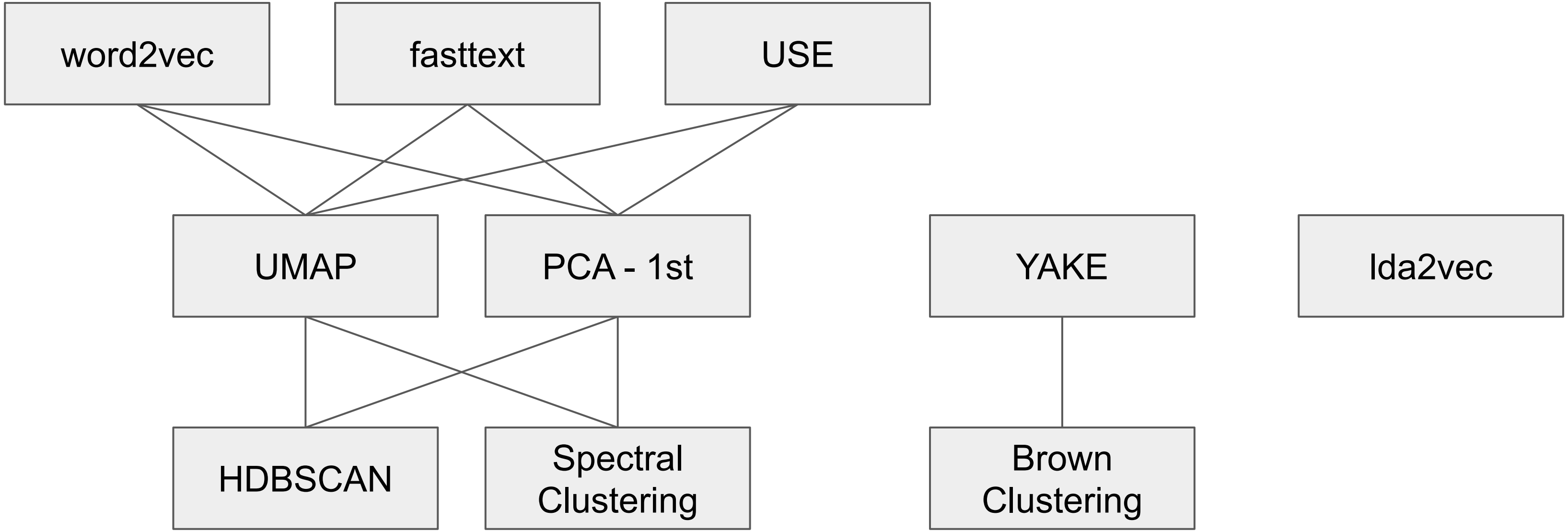
Clustering:

- Brown Clusters
  - Doesn’t require feature engineering; can take words directly
  - Hierarchical clusters (could be useful for visualization/exploration)
  - Can be actively updated (wouldn’t have to retrain)
- DBSCAN/H(ierarchical)DBSCAN
  - Could take embeddings
  - Clusters assumed to be of similar densities
- Spectral clustering
  - Doesn’t make assumptions about spatial distribution of data
  - In sklearn

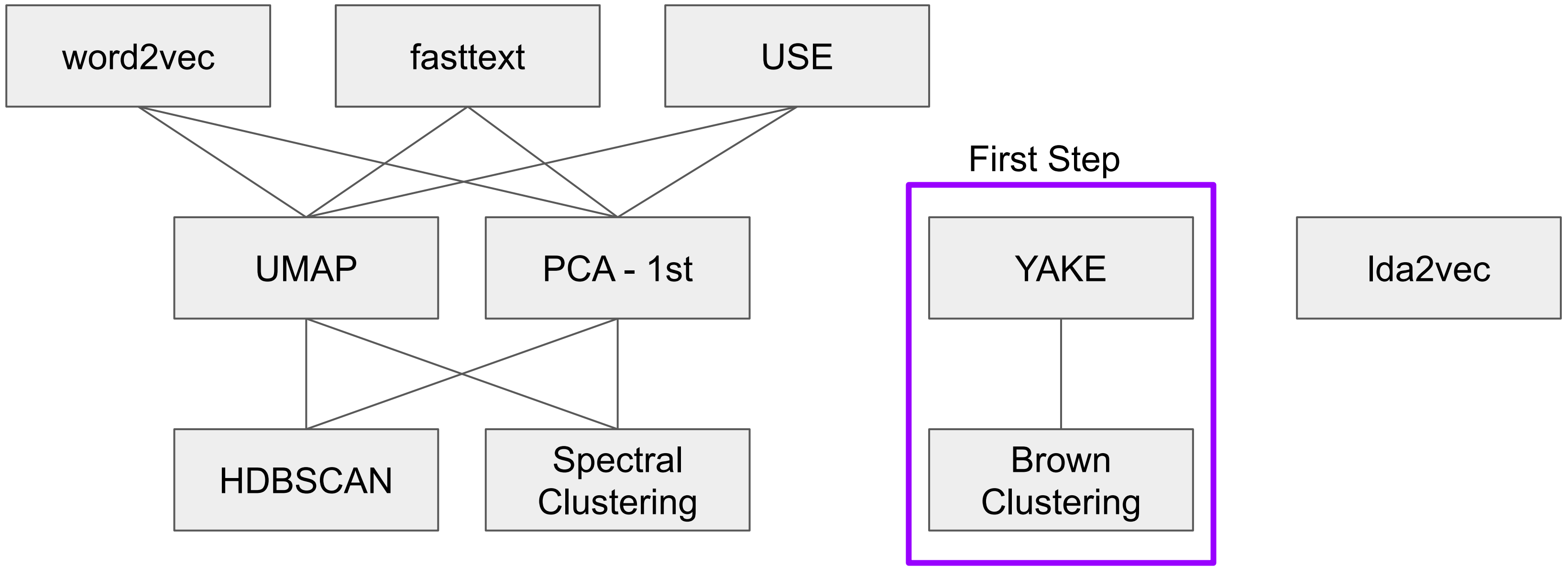
Clustering:

- Brown Clusters
  - Doesn’t require feature engineering; can take words directly
  - HIerarchical clusters (could be useful for visualization/exploration)
  - Can be actively updated (wouldn’t have to retrain)
- DBSCAN/H(ierarchical)DBSCAN
  - Could take embeddings
  - Clusters assumed to be of similar densities
- Spectral clustering
  - Doesn’t make assumptions about spatial distribution of data
  - In sklearn

Next stage: Experiments



Next stage: Experiments



YAKE -- Before

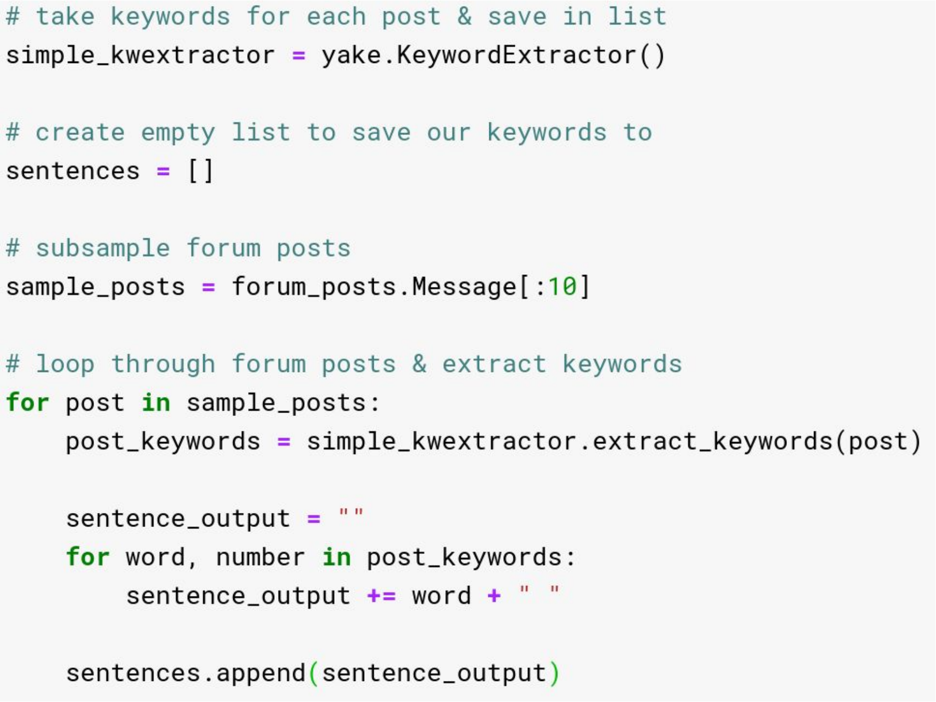
my background is from biology and, even if I have been doing bioinformatics for a few years now, I don't have enough knowledge of machine learning to solve this by myself: therefore, if someone is interested in making a two-people team with me, I would be glad to collaborate, provided that you explain the machine learning part to me.

In any case, since I am more interested in learning than in the prize of the competition, I will put here some ideas for everybody:

- the two sets of sequences represent coding sequences of two proteins; therefore, one thing to do is to translate them and compare the protein sequences. Even if two individuals have different DNA sequences for a gene, they can have the same protein sequences; and since only the protein is exposed to functional constraints, then it will be more interesting to see the differences in the protein sequences.
- analyzing k-mers doesn't seem very interesting to me. k-mers are usually used to identify regulatory motifs in DNA, which define when a gene is expressed, how, etc.. However, these signals usually are not inside the coding part of a gene sequence, but rather in the positions before or sorrounding the gene. So, the regulatory factors that you are looking with k-mers could be not included in the sequences given. For a similar reason, the GC content is not so informative.
- a possible approach would be to look at which sites are the most variable within the protein sequences.

YAKE -- After

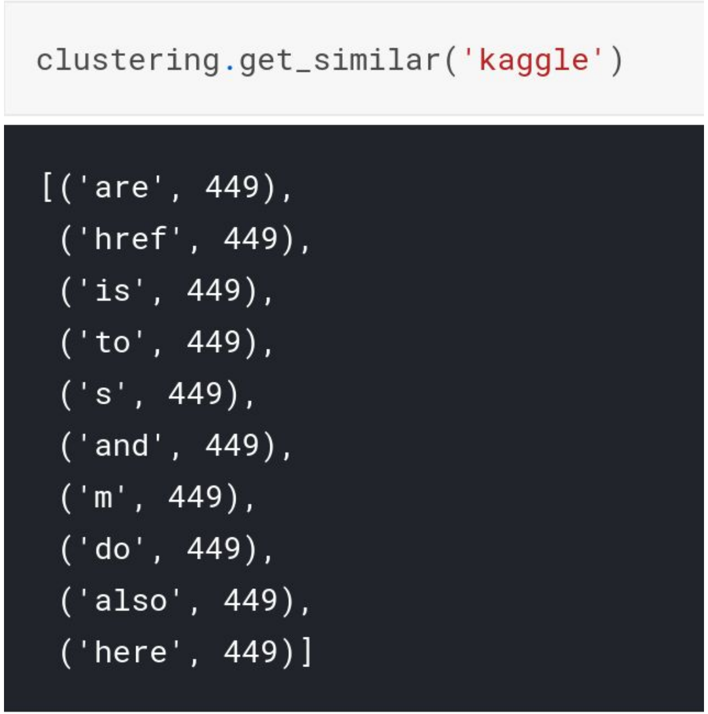
- **Keywords**:
  - machine learning part machine learning sequences represent coding represent coding sequences interested in making protein sequences glad to collaborate making a two-people two-people team learning part learning protein interested in learning sequences sequences represent dna sequences knowledge of machine explain the machine represent coding compare the protein
- One-fifth the length of the original post
- “Free” stopword removal
- Code: https://www.kaggle.com/rebeccaturner/yake-example/



```
# take keywords for each post & save in list
simple_kwextractor = yake.KeywordExtractor()

# create empty list to save our keywords to
sentences = []

# subsample forum posts
sample_posts = forum_posts.Message[:10]

# loop through forum posts & extract keywords
for post in sample_posts:
    post_keywords = simple_kwextractor.extract_keywords(post)

    sentence_output = ""
    for word, number in post_keywords:
        sentence_output += word + " "

    sentences.append(sentence_output)
```

Brown Clustering: Good news!

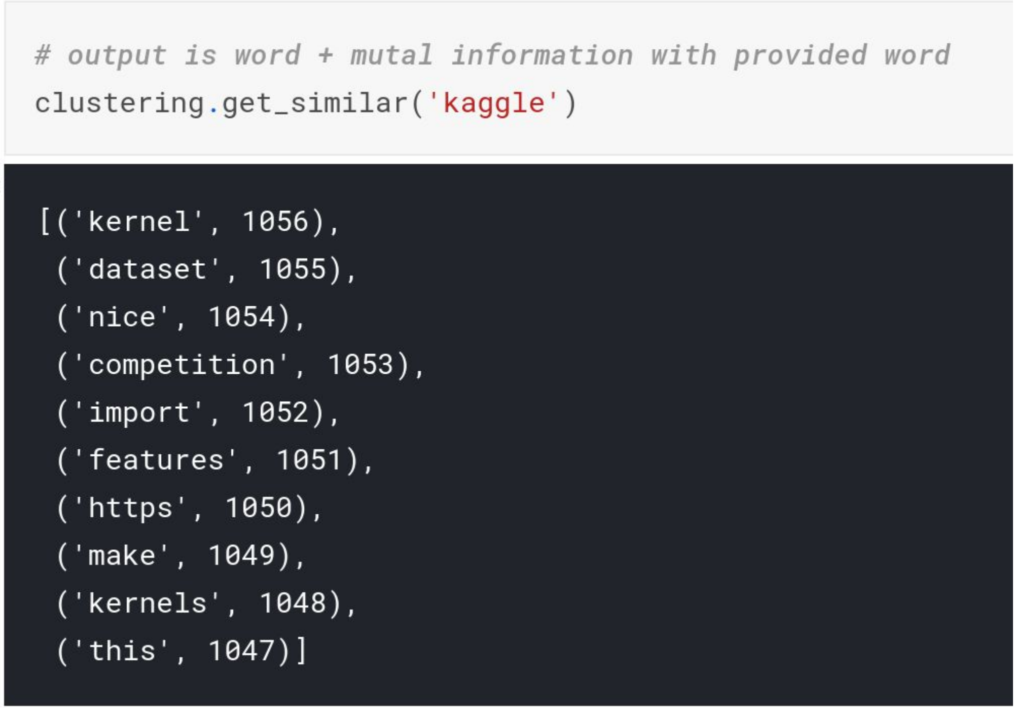
Without YAKE



```
clustering.get_similar('kaggle')
```

```
[('are', 449),
 ('href', 449),
 ('is', 449),
 ('to', 449),
 ('s', 449),
 ('and', 449),
 ('m', 449),
 ('do', 449),
 ('also', 449),
 ('here', 449)]
```

With YAKE

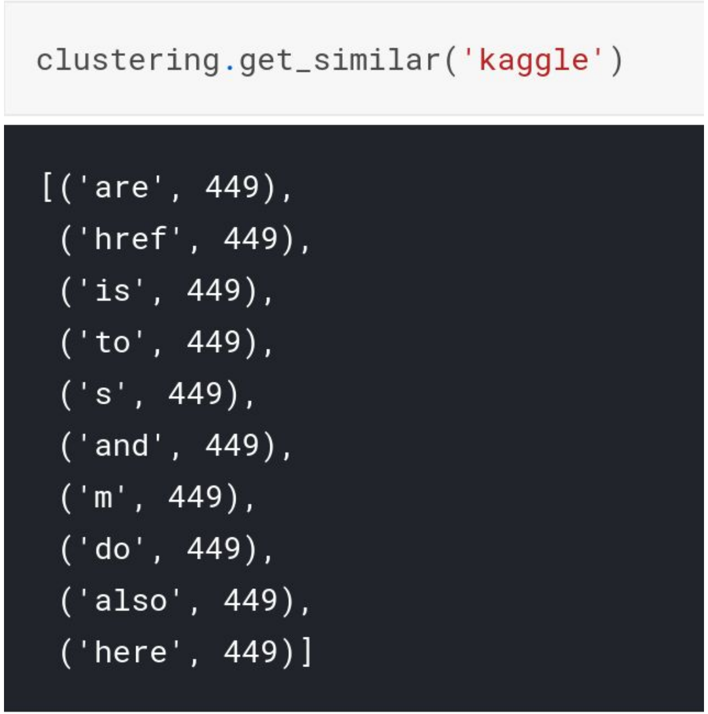


```
# output is word + mutal information with provided word
clustering.get_similar('kaggle')
```

```
[('kernel', 1056),
 ('dataset', 1055),
 ('nice', 1054),
 ('competition', 1053),
 ('import', 1052),
 ('features', 1051),
 ('https', 1050),
 ('make', 1049),
 ('kernels', 1048),
 ('this', 1047)]
```

Brown Clustering: Good news!

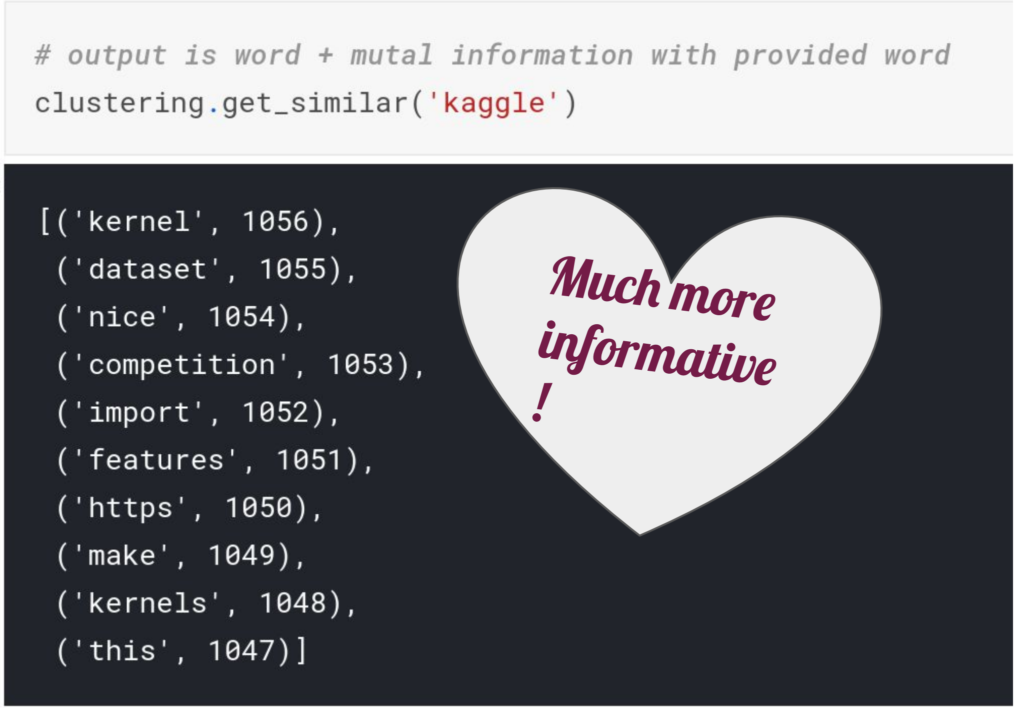
Without YAKE



```
clustering.get_similar('kaggle')
```

```
[('are', 449),
 ('href', 449),
 ('is', 449),
 ('to', 449),
 ('s', 449),
 ('and', 449),
 ('m', 449),
 ('do', 449),
 ('also', 449),
 ('here', 449)]
```

With YAKE



```
# output is word + mutal information with provided word
clustering.get_similar('kaggle')
```

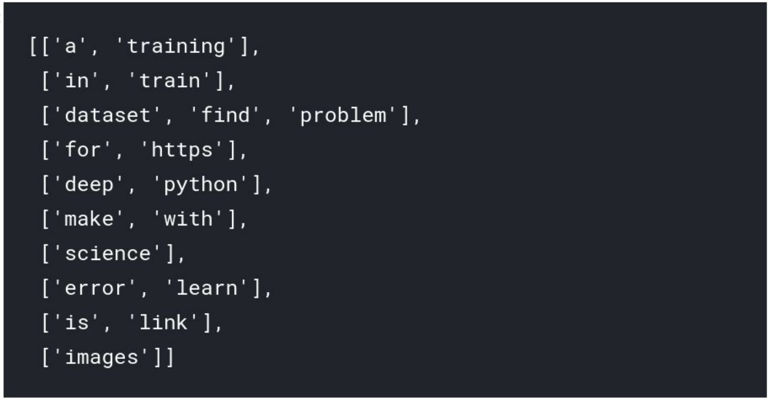
```
[('kernel', 1056),
 ('dataset', 1055),
 ('nice', 1054),
 ('competition', 1053),
 ('import', 1052),
 ('features', 1051),
 ('https', 1050),
 ('make', 1049),
 ('kernels', 1048),
 ('this', 1047)]
```

Much more informative !

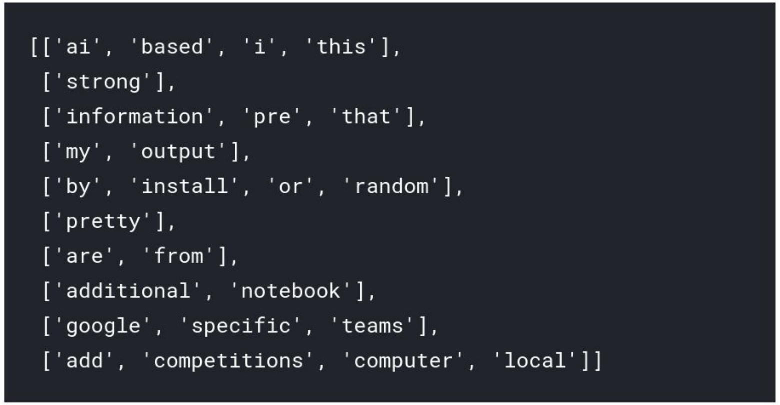
Brown Clustering: Bad news

- Library I was using missing some key methods (like returning clustering) so I needed to figure out on my own
- Very, very slow (author didn't recommend for lexicons above 5k so no surprise there)
- Final output was… not great even after extensive tuning

Back to the drawing board :'(

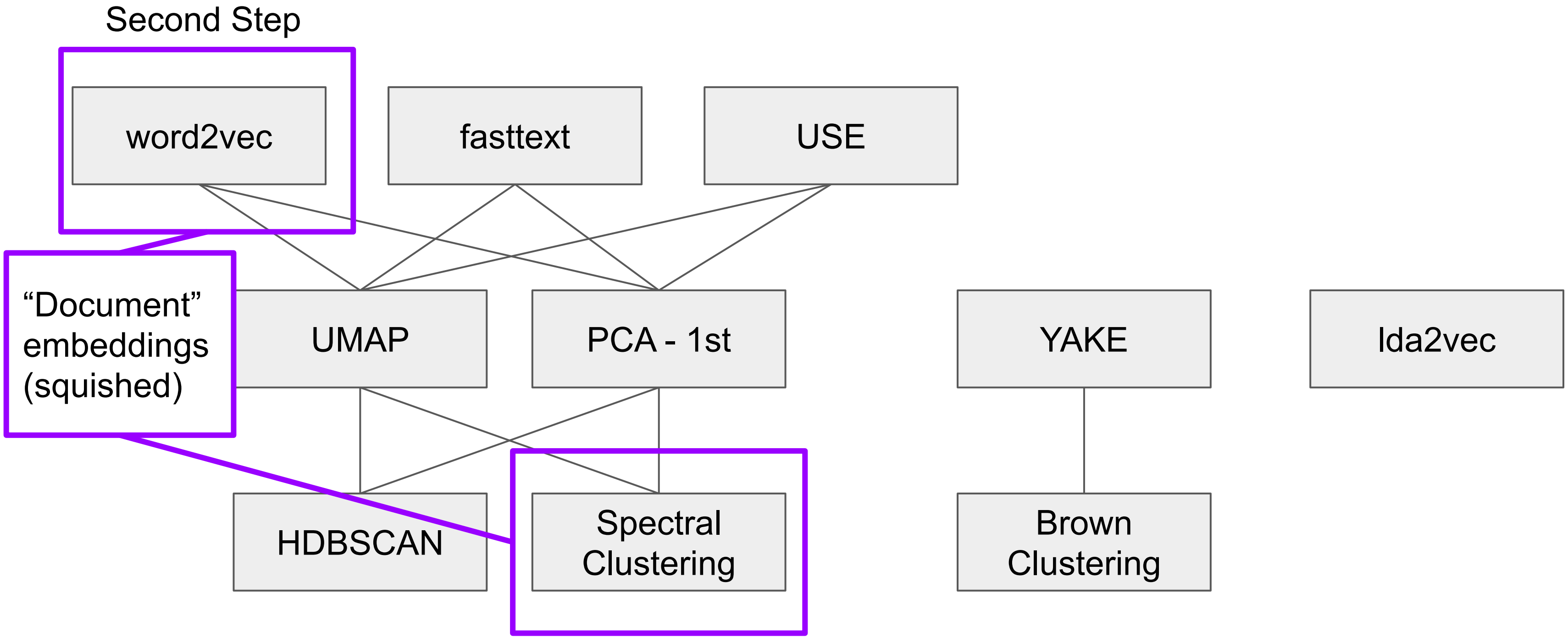


```
[['a', 'training'],
 ['in', 'train'],
 ['dataset', 'find', 'problem'],
 ['for', 'https'],
 ['deep', 'python'],
 ['make', 'with'],
 ['science'],
 ['error', 'learn'],
 ['is', 'link'],
 ['images']]
```



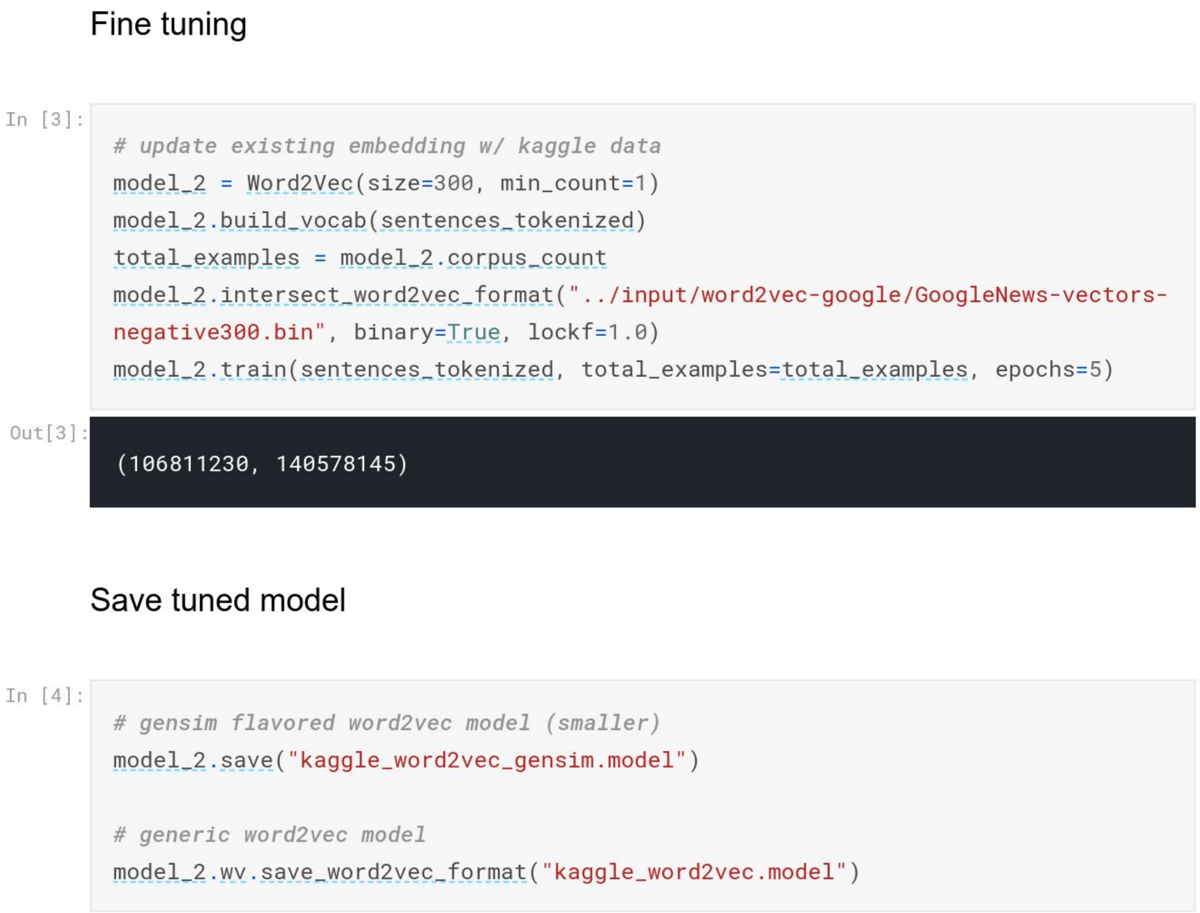
```
[['ai', 'based', 'i', 'this'],
 ['strong'],
 ['information', 'pre', 'that'],
 ['my', 'output'],
 ['by', 'install', 'or', 'random'],
 ['pretty'],
 ['are', 'from'],
 ['additional', 'notebook'],
 ['google', 'specific', 'teams'],
 ['add', 'competitions', 'computer', 'local']]
```

Next stage: Experiments



Word2Vec

- Used Word2Vec because they were tunable without retraining
- Tuned just once on corpus of whole Kaggle forums (August 2019)
- Tuned embeddings saved and used downstream



Fine tuning

In [3]:

```
# update existing embedding w/ kaggle data
model_2 = Word2Vec(size=300, min_count=1)
model_2.build_vocab(sentences_tokenized)
total_examples = model_2.corpus_count
model_2.intersect_word2vec_format("../input/word2vec-google/GoogleNews-vectors-negative300.bin", binary=True, lockf=1.0)
model_2.train(sentences_tokenized, total_examples=total_examples, epochs=5)
```

Out[3]:

```
(106811230, 140578145)
```

Save tuned model

In [4]:

```
# gensim flavored word2vec model (smaller)
model_2.save("kaggle_word2vec_gensim.model")

# generic word2vec model
model_2.wv.save_word2vec_format("kaggle_word2vec.model")
```

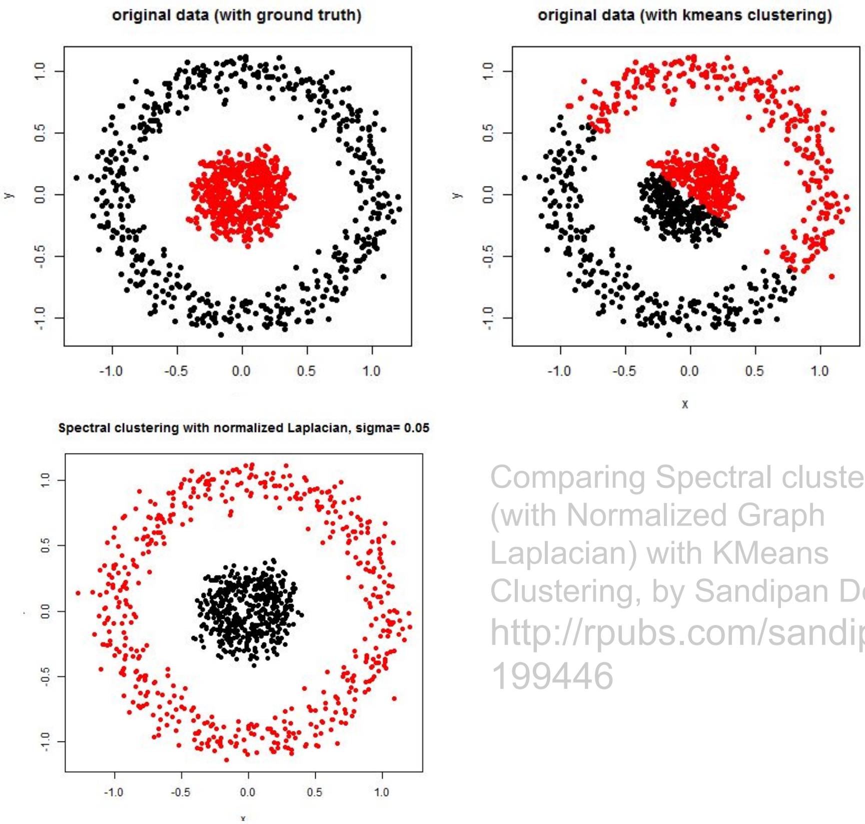
https://www.kaggle.com/rebeccaturner/fine-tuning-word2vec-2-0

“Document” embeddings

word	4	75	54	63
in	93	38	25	45
the	74	100	53	31
post	61	45	31	60
	MEAN =	MEAN =	MEAN =	MEAN =
“Document” embedding	58	64.5	40.75	49.75

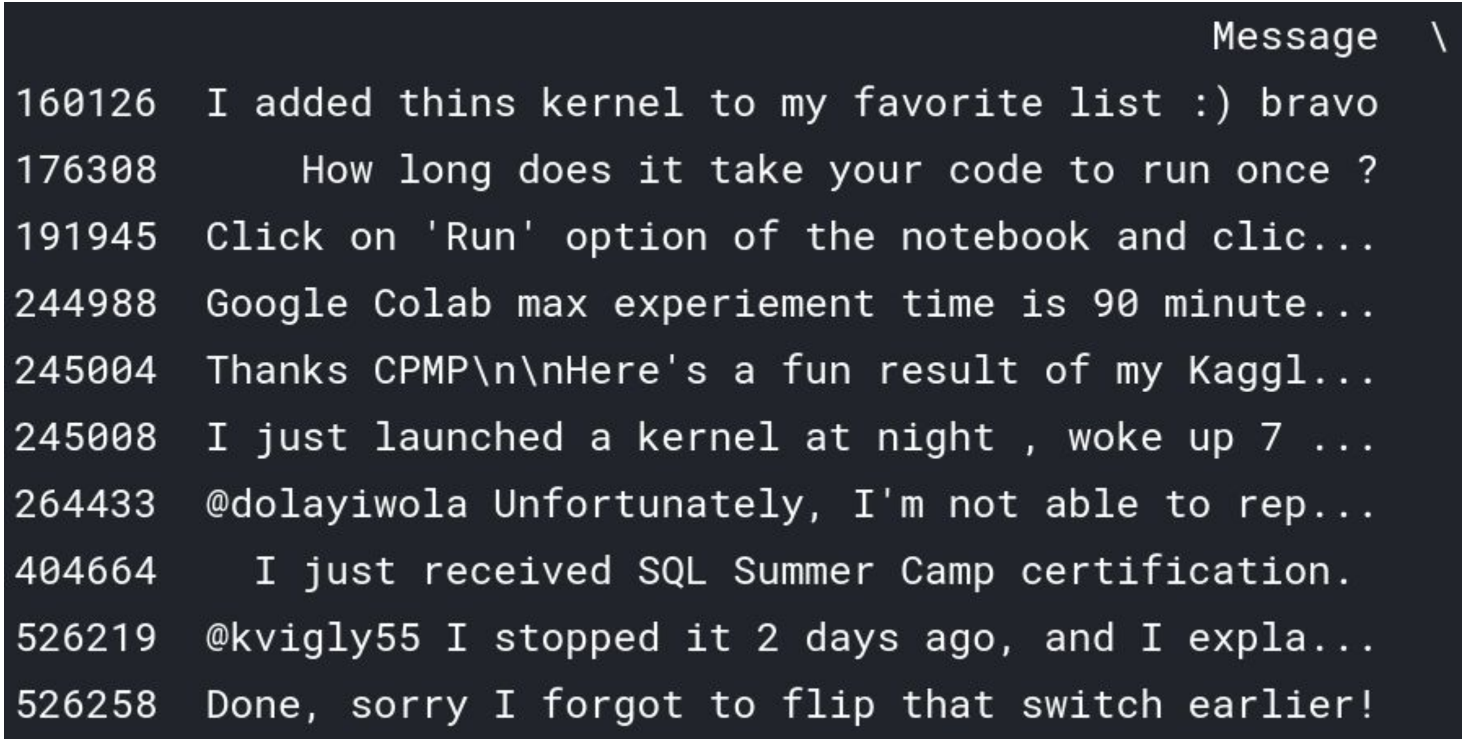
Spectral Clustering

- Simplest implementation: Connect all points with pairwise distance less than some pre-specified value
- Benefits:
  - Can model more complex decision regions
  - Doesn’t assume groups of similar size/shape
  - Don’t need to specify number of group ahead of time
  - Great for sparse data



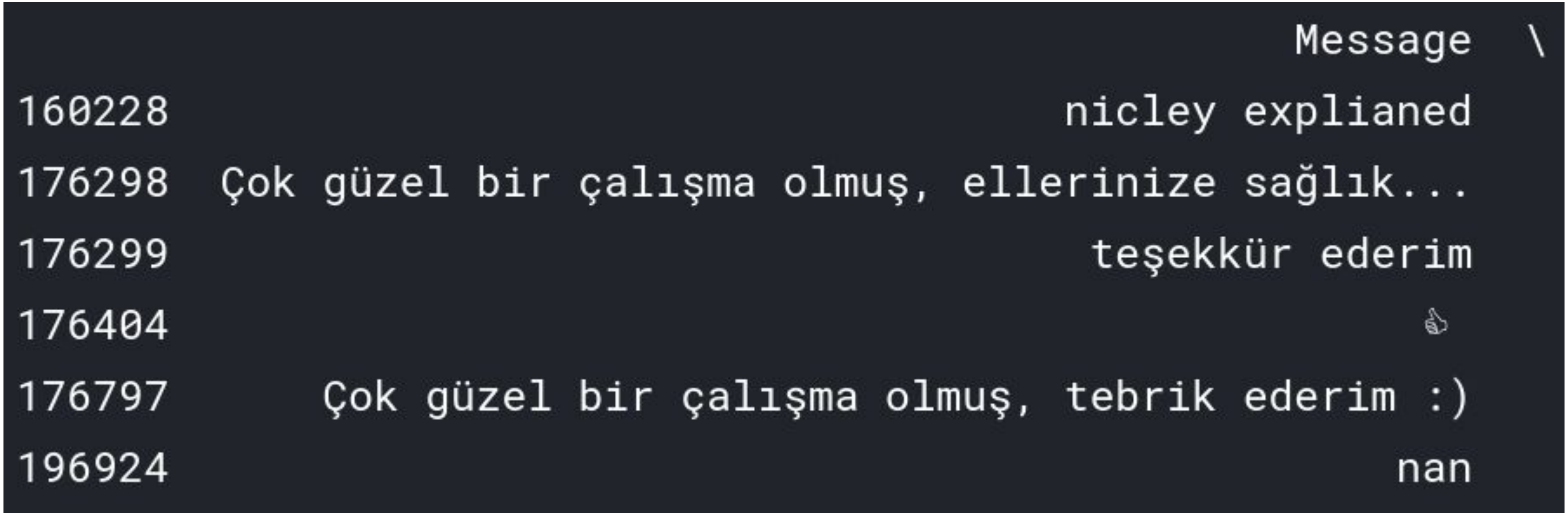
Comparing Spectral clustering (with Normalized Graph Laplacian) with KMeans Clustering, by Sandipan Dey http://rpubs.com/sandipan/199446

Clustering: Running Kernels (AKA Notebooks)



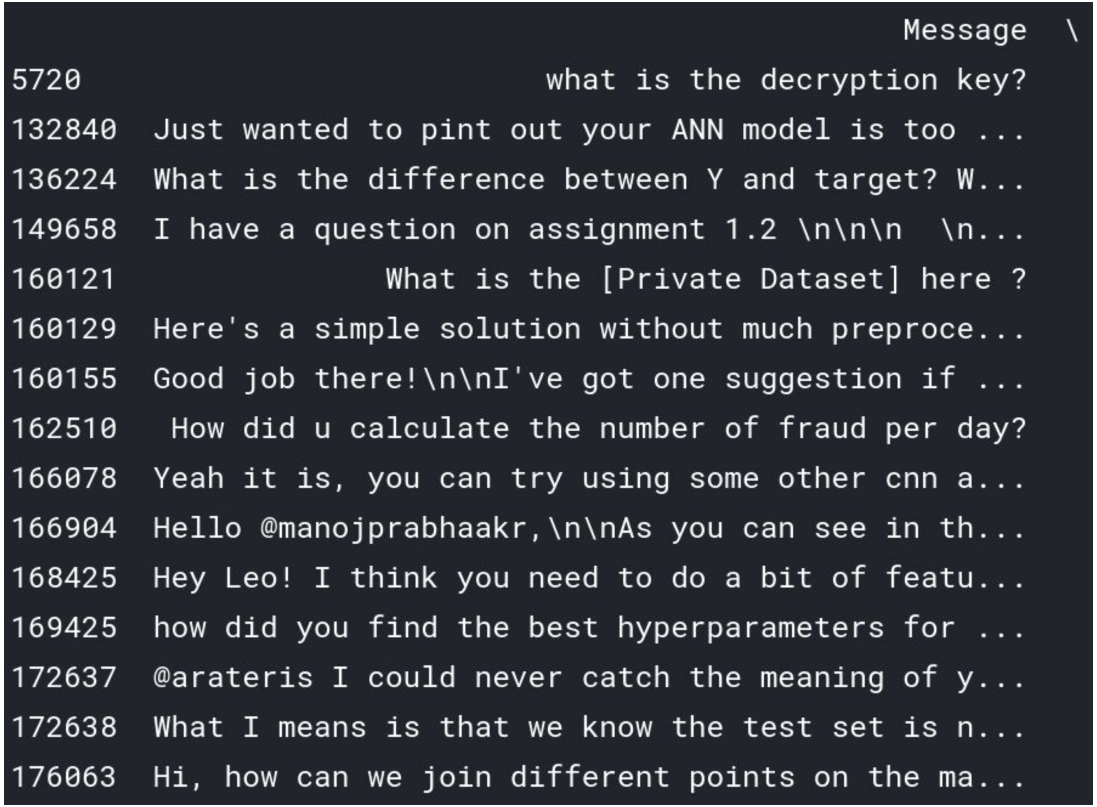
```
                                                   Message  \
160126  I added thins kernel to my favorite list :) bravo
176308      How long does it take your code to run once ?
191945  Click on 'Run' option of the notebook and clic...
244988  Google Colab max experiement time is 90 minute...
245004  Thanks CPMP\n\nHere's a fun result of my Kaggl...
245008  I just launched a kernel at night , woke up 7 ...
264433  @dolayiwola Unfortunately, I'm not able to rep...
404664    I just received SQL Summer Camp certification.
526219  @kvigly55 I stopped it 2 days ago, and I expla...
526258  Done, sorry I forgot to flip that switch earlier!
```

Clustering: Turkish (Thanks/Congrats)



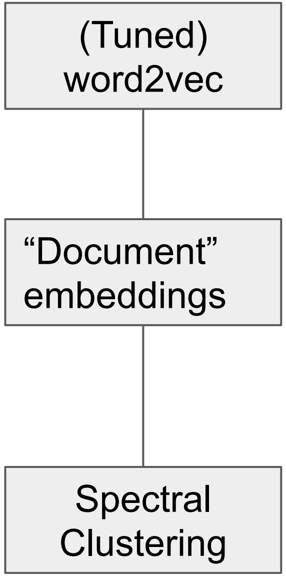
```
                                                  Message  \
160228                                   nicley explianed
176298  Çok güzel bir çalışma olmuş, ellerinize sağlık...
176299                                    teşekkür ederim
176404                                                 👍
176797      Çok güzel bir çalışma olmuş, tebrik ederim :)
196924                                                nan
```

Clustering: ML Questions & Answers



```
                                                 Message  \
5720                         what is the decryption key?
132840  Just wanted to pint out your ANN model is too ...
136224  What is the difference between Y and target? W...
149658  I have a question on assignment 1.2 \n\n\n  \n...
160121              What is the [Private Dataset] here ?
160129  Here's a simple solution without much preproce...
160155  Good job there!\n\nI've got one suggestion if ...
162510   How did u calculate the number of fraud per day?
166078  Yeah it is, you can try using some other cnn a...
166904  Hello @manojprabhaakr,\n\nAs you can see in th...
168425  Hey Leo! I think you need to do a bit of featu...
169425  how did you find the best hyperparameters for ...
172637  @arateris I could never catch the meaning of y...
172638  What I means is that we know the test set is n...
176063  Hi, how can we join different points on the ma...
```

Current Unsupervised Pipeline



(Tuned) word2vec

"Document" embeddings

Spectral Clustering

All code is public & open source (Apache 2.0):

- Fine tuning: https://www.kaggle.com/rtatman/fine-tuning-word2vec/
- Full pipeline (including some work on summarization): https://www.kaggle.com/rtatman/forum-post-embeddings-clustering
- Live coding recordings: https://www.youtube.com/playlist?list=PLqFaTIg4myu9f21aM1POYVeoaHbFf1hMc

Thanks! Questions?