Similarity Inspection Using Clustering Approach

By

Mahmoud Yahia Ahmed

Abdulrahman Mohamed Abdelsalam

Mahmoud Maged Mohamed

Umar Mohamed Ibrahim

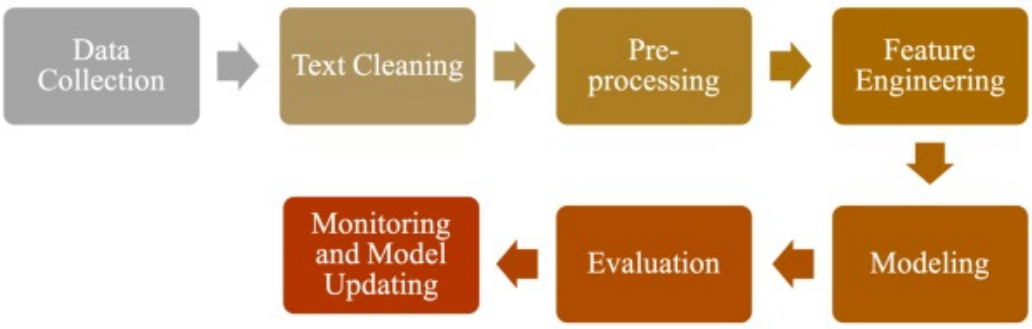
Group
DSA_202101_12
19 June 2022

Goal

The overall aim is to produce similarity inspections following the clustering approach and compare; analyze the pros and cons of algorithms, and generate and communicate the insights.

Abstract

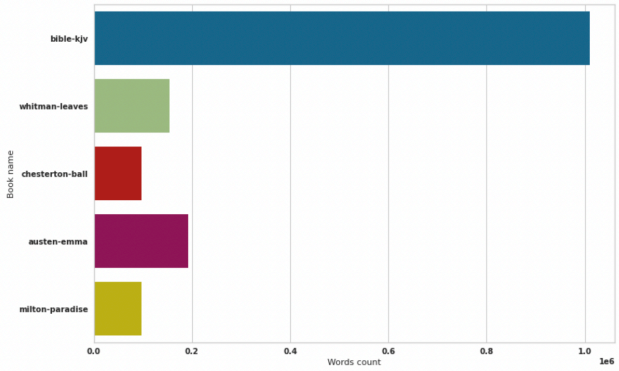
Recently, the rise of big data and natural language processing algorithms has gained a huge amount of interest from competing technology companies. Text Clustering is one of the problems solved by NLP algorithms. Text clustering refers to the process of unsupervised learning of specified text. This report describes the various techniques of text clustering, including text representation, feature engineering, and clustering algorithms, and draws the basic ideas, advantages, and disadvantages of several current



Dataset

five books were selected from Gutenberg's digital library, and all of them have different genre. Books Names e.g

- bible-kjv (Religious)
- whitman-leaves (Drama)
- chesterton-ball (Children)
- austen-emma (Fiction)
- milton-paradise (Poetry)

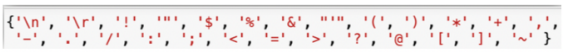


The illustration shows a huge variation in the word counts per book, and because that data is imbalanced, 200 random paragraphs were chosen from each book, and each paragraph has 150 words, so our corpus will contain 1K paragraph of 5 books.

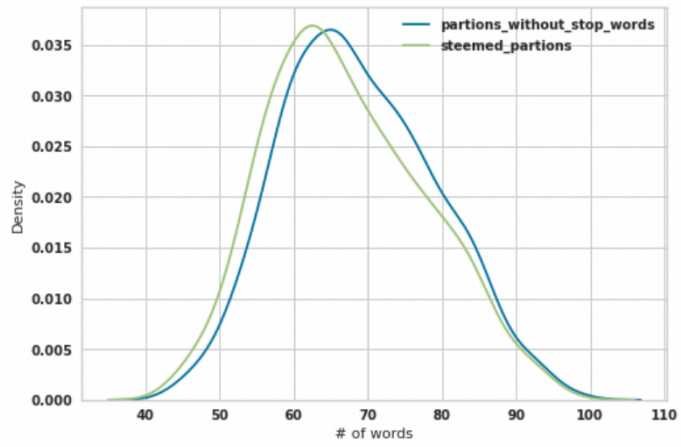
Data Cleaning

We cleaned the words from symbols and removing any not needed numbers and punctuations, lastly we removed any stop words from text. Then we performed partitioning, by taking random 200 partitions from each book, each partition consists of 150 words. Then we performed labelling and indexing and creating a Dataframe. Then we performed Exploratory Data Analysis (EDA) and showed the word cloud to know and take insights of the most repeated words and terminologies in each book as it will definitely affect our model through the different transformation techniques.

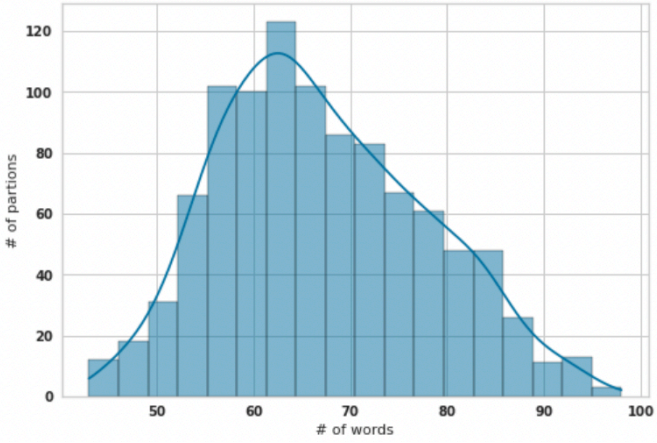
A. Clean Special Characters.



B. Remove Stop Words and Stemming.



the curve shows how removing stop words and stemming the data affect the words count distribution.



the histogram shows the frequencies after removing stop words and stemming.

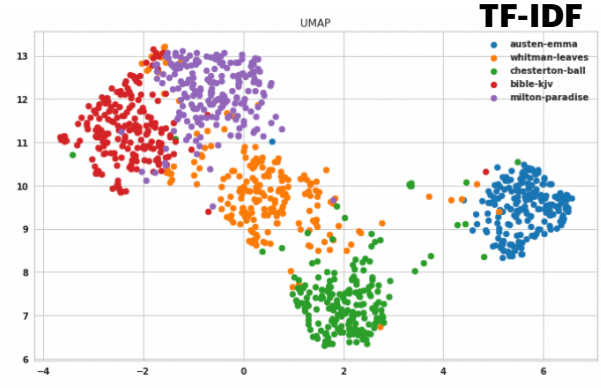
Transformation (Embedding)

Multiple embedding techniques were used at that phase, some of them depend on word occurrence e.g. "BOW", "Tf-IDF", "N-gram", in addition to some techniques that maintain the meaning e.g. "Glove", "Word2Vec" and "FastText", but the last three techniques we used pre-trained models to apply transfer-learning and get the contextual embedding for each word instead of naive occurrence-based one.

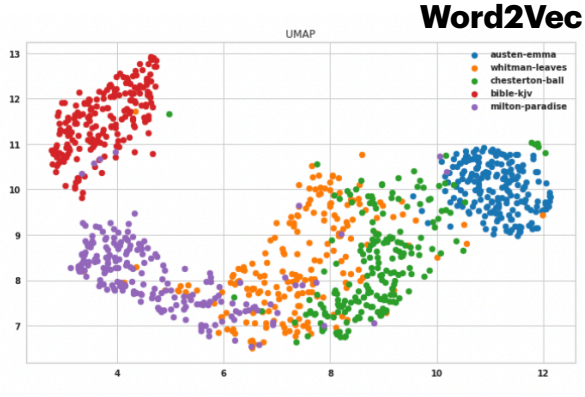
Clustering Separability checking

It's crucial to make sure the data separability nature, just to make sure of the clustering step feasibility, we have checked the Clustering Separability visually between each Embedding technique and 4 dimensionality reduction algorithms, and our conclusion is that **U-MAP** is the best in separating the clusters with the occurrence of embedding techniques, but LDA is the best for contextual embedding techniques, but because we need to deal with the absent y_true case (unknown) as the most real case, so we will apply **U-MAP** for all of them, also to fit the memory.

TF-IDF



Word2Vec



Modeling

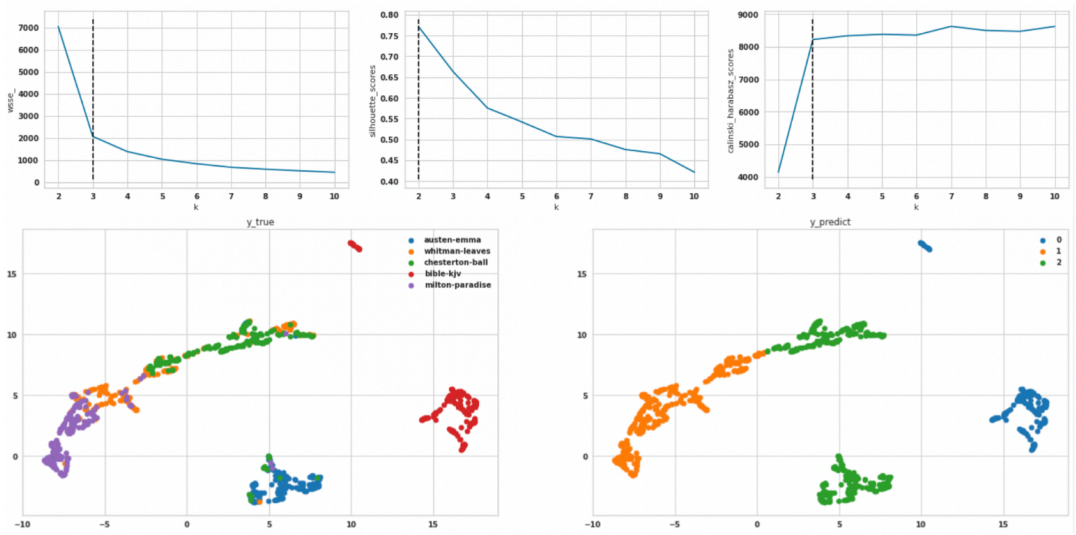
This elaborate the comparing the capabilities of three different models in data separation depend on the separation nature following unsupervised approach, there paradigms were

- K-means (non parametric).
- Estimation-Maximization EM, Generative model (parametric).
- Agglomerative Clustering, Hierarchical.

K-means

	name	vec_shape	wsse	best_k_elbow	silhouette	best_k_silhouette	calinski	best_k_calinski
0	K-mean-bow_umap	(1000, 10)	762.862366	5	0.500059	2	1042.547322	2
1	K-mean-tfidf_umap	(1000, 10)	850.376099	4	0.576124	2	3727.280846	5
2	K-mean-bow-n-gram_umap	(1000, 10)	673.701050	5	0.282630	2	407.725894	2
3	K-mean-tfidf-n-gram_umap	(1000, 10)	810.976440	4	0.404289	2	1186.523205	3
4	K-mean-glove_umap	(1000, 10)	923.145996	4	0.633800	2	3966.040685	4
5	K-mean-word2vec_umap	(1000, 10)	991.435547	4	0.603739	2	3856.803810	4
6	K-mean-fastTxt_umap	(1000, 10)	2078.271484	3	0.771957	2	8218.903619	3

FastText Embedding with U-MAP dimensionality reduction got the highest **silhouette**, **calinski** and lowest **wsse**, and the majority voting is **K=3**, also when k=3 the silhouette will not affect too much.

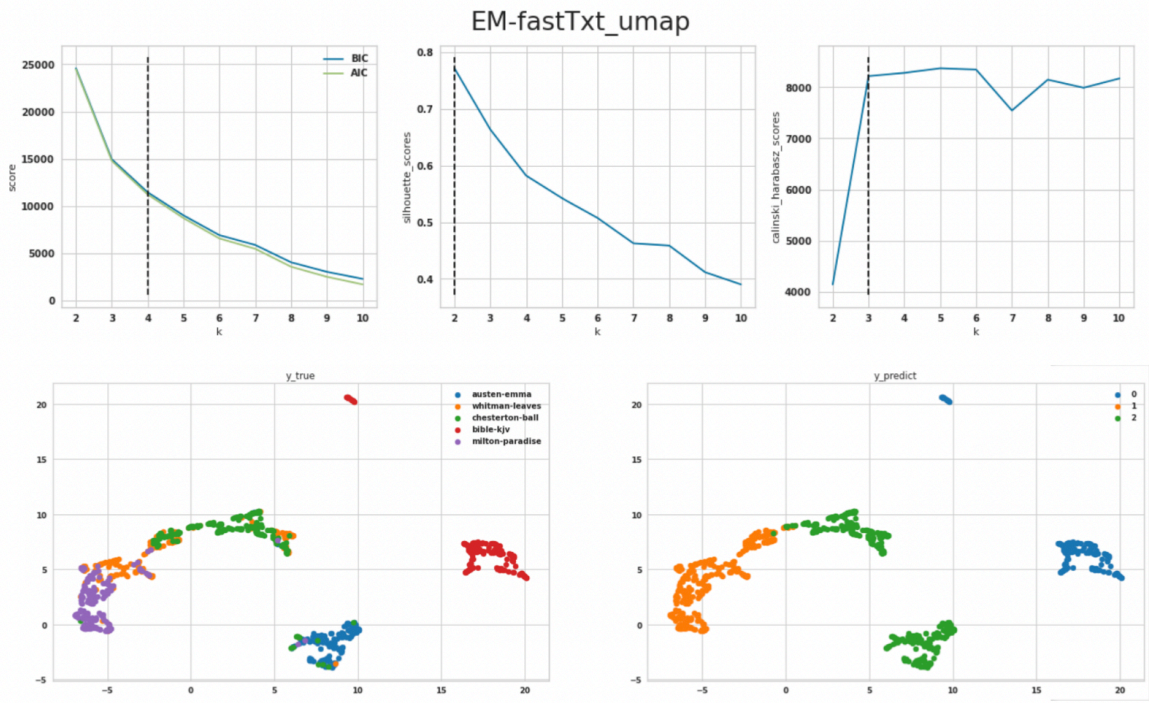


U-MAP applied here to reduce dimensions to 2d just visualization.

Estimation-Maximization

	name	vec_shape	bic_score	best_k_bic	aic_score	best_k_aic	silhouette	best_k_silhouette	calinski	best_k_calinski
0	EM-bow_umap	(1000, 10)	4966.411379	5	4676.853818	5	0.503459	2	1037.948402	2
1	EM-tfidf_umap	(1000, 10)	3406.235581	5	3116.678020	5	0.567237	2	3727.140743	5
2	EM-bow-n-gram_umap	(1000, 10)	4791.563754	5	4502.006192	5	0.284044	2	406.957435	2
3	EM-tfidf-n-gram_umap	(1000, 10)	5773.827906	4	5543.163408	4	0.401408	2	1171.137280	3
4	EM-glove_umap	(1000, 10)	6874.854046	4	6644.189548	4	0.636421	2	3799.021966	4
5	EM-word2vec_umap	(1000, 10)	7570.762121	4	7340.097623	4	0.604721	2	3719.559882	4
6	EM-fastTxt_umap	(1000, 10)	11466.142222	4	11235.477724	4	0.771957	2	8218.903619	3

FastText Embedding with LDA dimensionality reduction got the highest **silhouette**, **calinski** and lowest **BIC**, and lowest **AIC**, and according to Elbow charts, **K=3** is the most sensible.

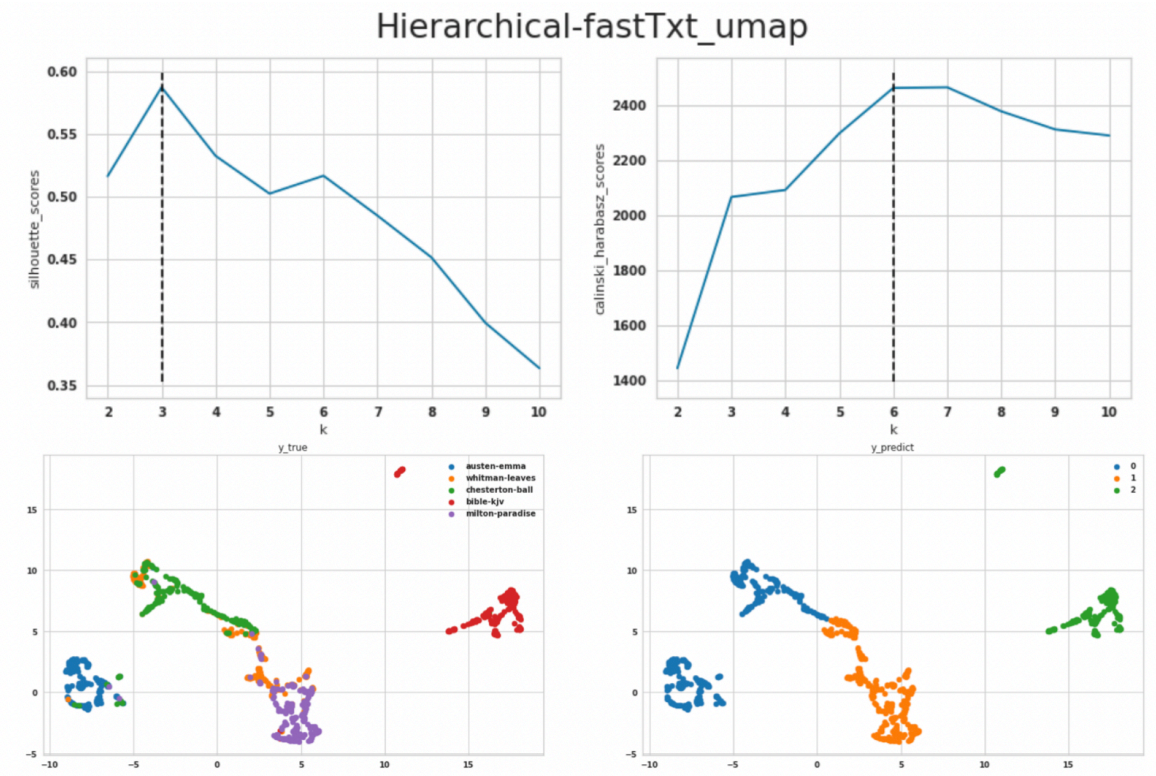


U-MAP applied here to reduce dimensions to 2d just visualization.

Agglomerative Clustering-Hierarchical

	name	vec_shape	silhouette	best_k_silhouette	calinski	best_k_calinski
0	Hierarchical-bow_umap	(1000, 10)	0.393299	2	655.438782	2
1	Hierarchical-tfidf_umap	(1000, 10)	0.559137	2	1990.313745	3
2	Hierarchical-bow-n-gram_umap	(1000, 10)	0.263297	5	332.026312	2
3	Hierarchical-tfidf-n-gram_umap	(1000, 10)	0.387339	2	778.559201	3
4	Hierarchical-glove_umap	(1000, 10)	0.504256	2	1221.472043	3
5	Hierarchical-word2vec_umap	(1000, 10)	0.530212	2	1577.989317	3
6	Hierarchical-fastTxt_umap	(1000, 10)	0.587057	3	2462.740998	6

FastText Embedding with UMAP dimensionality reduction got the highest **silhouette**, **calinski,** k=3.

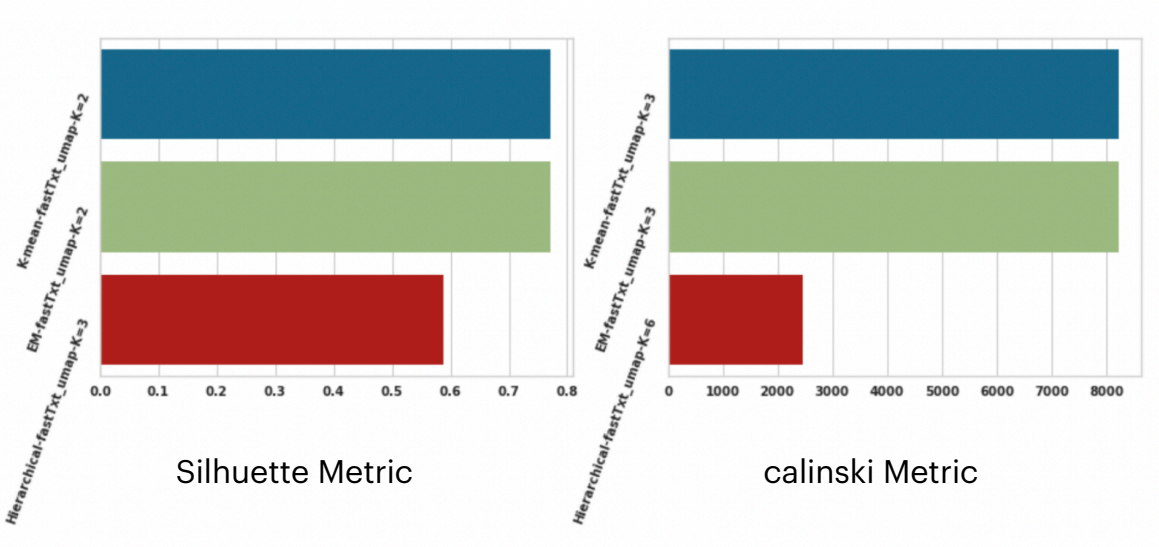


U-MAP applied here to reduce dimensions to 2d just visualization.

Best Model

Choose the winner model, that success to cluster the data with highest separability and will chosen based.

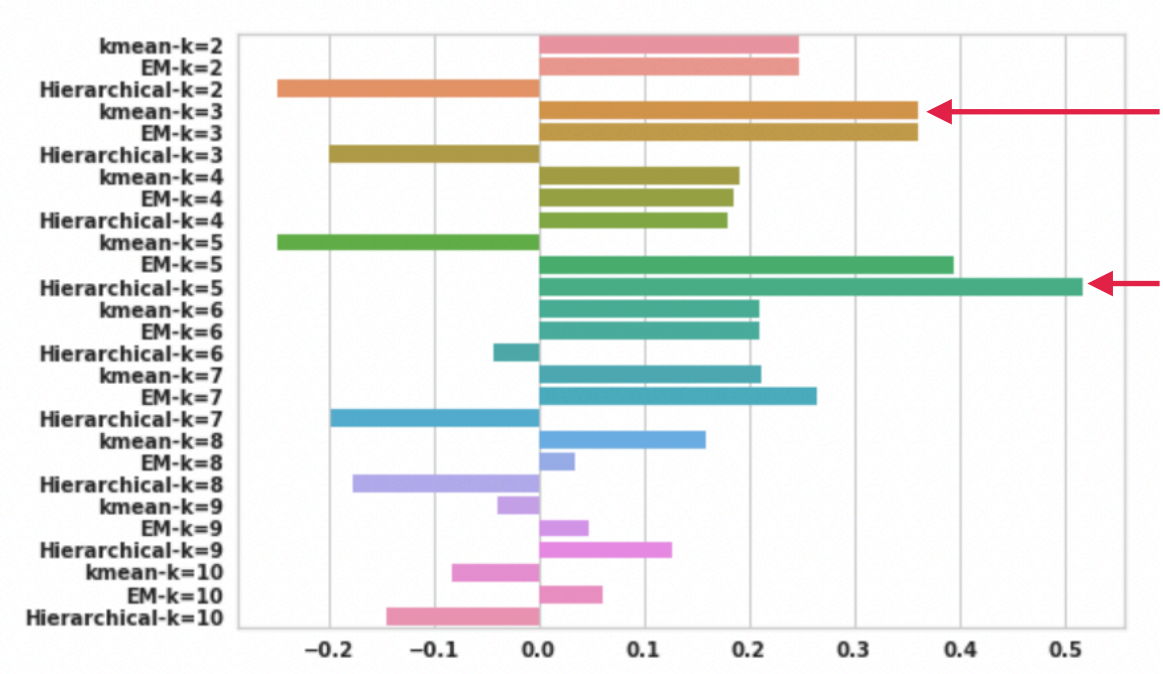
1. maximized the number of clusters (k=3).
2. maximized the silhouette_score & calinski_harabasz_score.

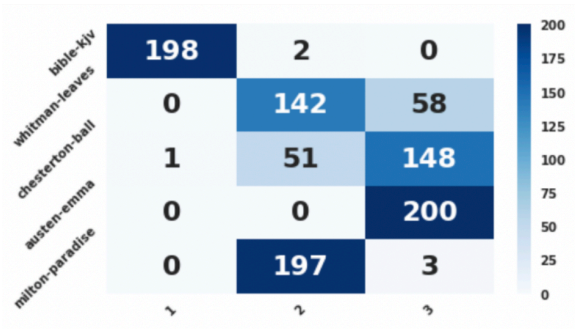


According to the results, we will choose K-means as best model, because it has the same silhouette and highest calinski, and we will choose K=3 as the best, because the silhouette will not decrease too much.

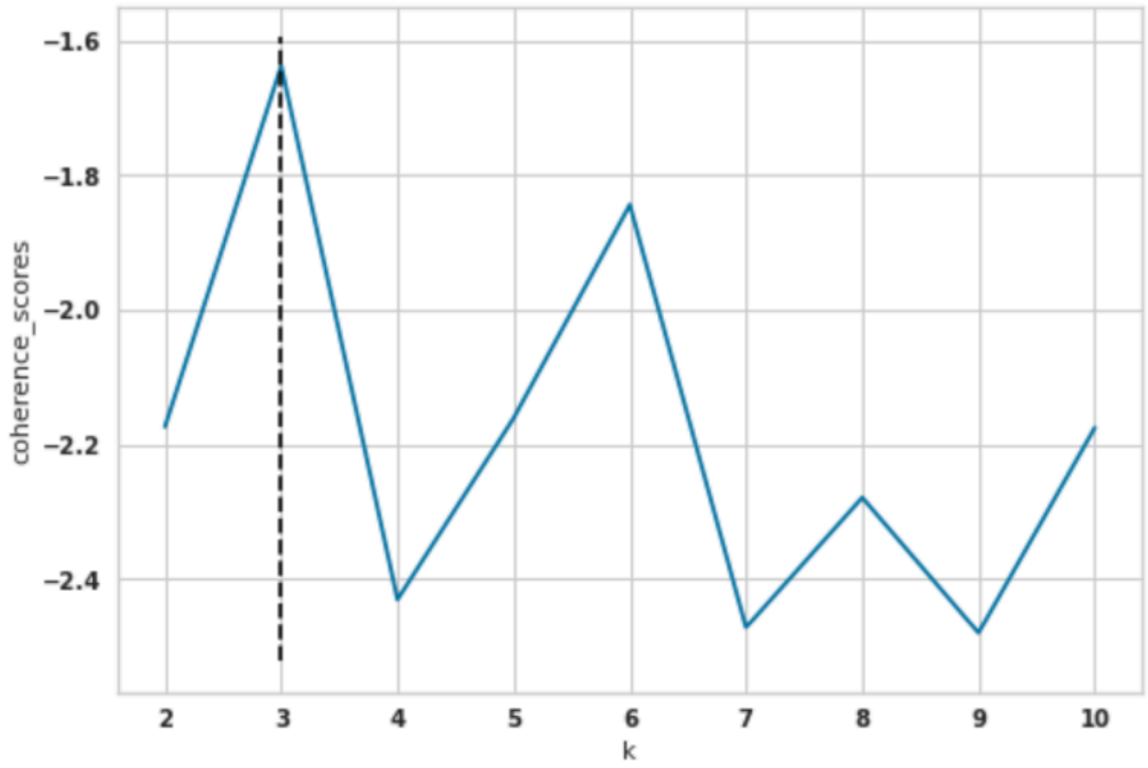
Kappa & coherence

K-means with 3 achieved 0.36 with Kappa measurement, but the highest Kappa achieved by **Hierarchical** with k=5, but we still dealing with k-means K=3.





From the confusion matrix above, there are some clusters that contain different books, which means those books were written in the same context because the FastText embedding was used and it's a contextual embedding, and the authors use the same lingo or same latent meaning to express their ideas. E.g clusters 1 and 2, but cluster 0, didn't overlap because it has a religious book. Another consideration, we already know that we have 5 books with different genres, and the number of clusters should be 5, and best k was just 3, according to model perspective, and we can't claim that this observation is wrong because here the cluster using the latent similarity which indistinct for humans, so that may be correct.

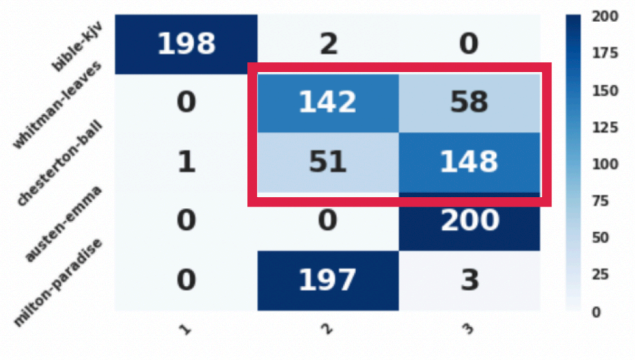


The best K-topics according to U_mass Coherence_score is 3.

Error Analysis

This figure shows that book bible-kjv is bound in cluster one, milton- paradise and austen-emma are bounded in cluster 2, 3 respectively, but for books whitman-leaves and chesterton-ball they overlapped with the cluster 2, 3.

The red box elaborate the overlapping that exist and that's maybe because of the authors' contextual writing, or they have used the same words exactly or words with similar which will be close to each other in the Embedding space.



	word	whitman-leaves	austen-emma	chesterton-ball
0	man	117	54	181
1	think	43	125	72
2	come	68	78	82
3	know	48	100	79
4	like	36	36	129
5	look	41	66	78
6	thing	28	87	60
7	day	92	61	15
8	good	44	83	38
9	little	25	82	47

Top 10 common words in Cluster-3

	word	milton-paradise	whitman-leaves	chesterton-ball
0	man	87	117	181
1	shall	120	78	21
2	like	40	36	129
3	know	59	48	79
4	come	32	68	82
5	day	73	92	15
6	long	53	59	44
7	good	69	44	38
8	look	27	41	78
9	great	63	53	29

Top 10 common words in Cluster-2

Those words confuse affect the accuracy and this is due to a common existence words within different clusters.

Conclusion

Assessment the Clustering Separability is very important, and there are many measurements that assess that separation success, some of them depend on y truth existence and some of them following statistical approaches, and which one to use it depends on the case and the problem, but rely on just one is not good think, you decision should taken based on combinations of them, because each one provide a specific curial measurement, also the embedding way affect the results significantly, the latent information exist in the embedding vector define how the similarity will be, and how each cluster contains with similar points, it also provide the sensible reasoning.

Thanks.