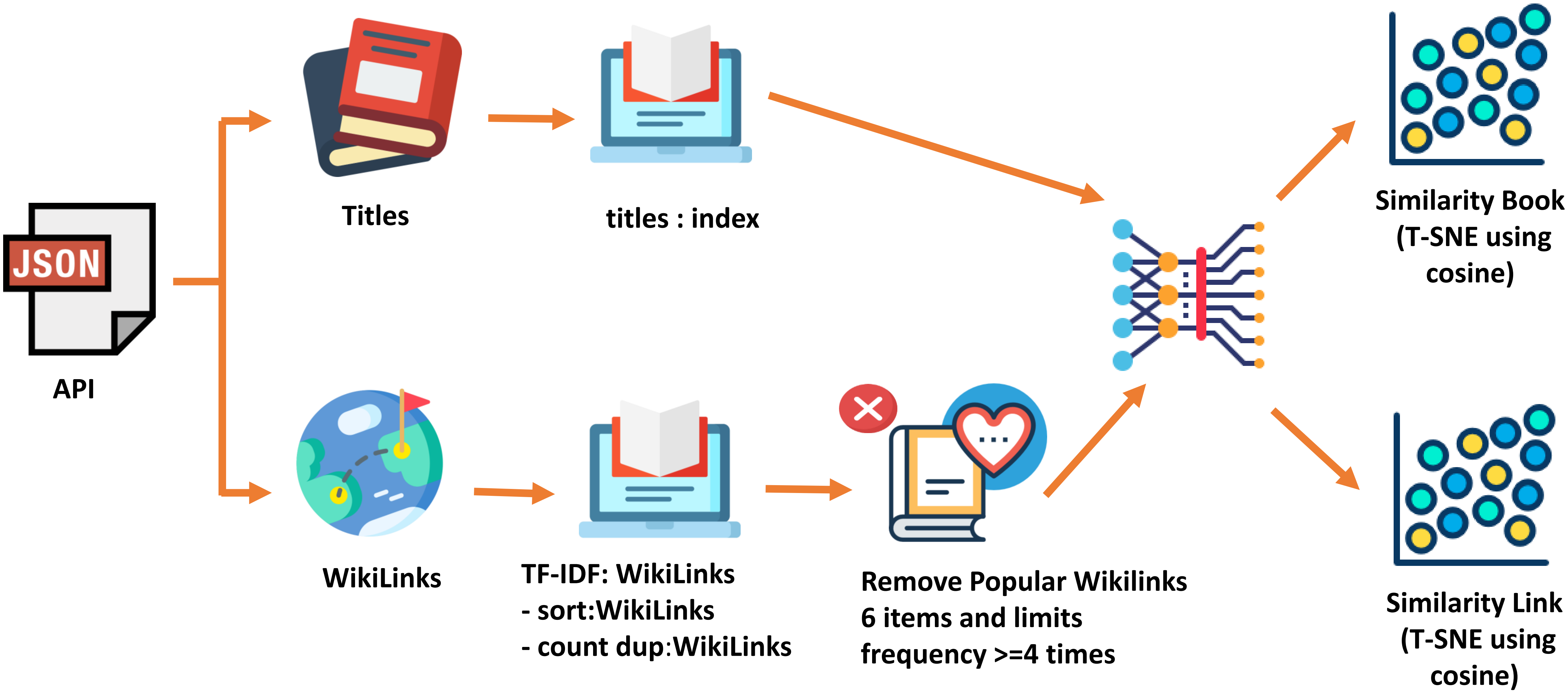
BOOK RECOMMENDATION ON WIKIPEDIA BY WARUNEE SERMPANICHAKIJ

Warunee Sermpanichakij 625162020007

EXPLORE DATA



DATA CLEANING

```
('Limonov (novel)',
 {'name': 'Limonov',
  'author': 'Emmanuel Carrère',
  'translator': 'John Lambert',
  'country': 'France',
  'language': 'French',
  'publisher': 'P.O.L.',
  'pub_date': '2011',
  'english_pub_date': '2014',
  'pages': '488',
  'isbn': '978-2-8180-1405-9'},
 ['Emmanuel Carrère',
  'biographical novel',
  'Emmanuel Carrère',
  'Eduard Limonov',
  'Prix de la langue française'],
 ['http://www.lefigaro.fr/flash-actu/2011/10/05/97001-20111005FILWWW00615-le-prix-de-la-lang
ue-francaise-a-e-carrere.php',
  'http://www.lexpress.fr/culture/livre/emmanuel-carrere-prix-renaudot-2011_1046819.html',
  'http://limonow.de/carrere/index.html',
  'http://www.tout-sur-limonov.fr/222318809'],
 ['http://www.lefigaro.fr/flash-actu/2011/10/05/97001-20111005FILWWW00615-le-prix-de-la-lang
ue-francaise-a-e-carrere.php',
  'http://www.lexpress.fr/culture/livre/emmanuel-carrere-prix-renaudot-2011_1046819.html',
  'http://limonow.de/carrere/index.html',
  'http://www.tout-sur-limonov.fr/222318809'],
 '2018-08-18T02:03:21Z',
 1437)
```

```
Using TensorFlow backend.

Downloading data from https://raw.githubusercontent.com/WillKoehrsen/wikipedia-data-science/
master/data/found_books_filtered.ndjson
58933248/58925764 [==============================] - 1s 0us/step
Found 37020 books.
```

```python
book_index = {book[0]: idx for idx, book in enumerate(books)}
index_book = {idx: book for book, idx in book_index.items()}

book_index['Anna Karenina']
index_book[22494]
```

```
22494

'Anna Karenina'
```

```
There are 297624 unique wikilinks.

[('paperback', 8740),
 ('hardcover', 8648),
 ('wikipedia:wikiproject books', 6043),
 ('wikipedia:wikiproject novels', 6016),
 ('science fiction', 5665),
 ('english language', 4248),
 ('united states', 3063),
 ('novel', 2983),
 ('the new york times', 2742),
 ('fantasy', 2003)]
```

Count wikilink

DATA CLEANING REMOVE POPULAR

```
[('Hardcover', 7489),
 ('Paperback', 7311),
 ('Wikipedia:WikiProject Books', 6043),
 ('Wikipedia:WikiProject Novels', 6015),
 ('English language', 4185),
 ('United States', 3060),
 ('Science fiction', 3030),
 ('The New York Times', 2727),
 ('science fiction', 2502),
 ('novel', 1979)]
```

lower

```
There are 297624 unique wikilinks.

[('paperback', 8740),
 ('hardcover', 8648),
 ('wikipedia:wikiproject books', 6043),
 ('wikipedia:wikiproject novels', 6016),
 ('science fiction', 5665),
 ('english language', 4248),
 ('united states', 3063),
 ('novel', 2983),
 ('the new york times', 2742),
 ('fantasy', 2003)]
```

- Remove Most Popular Wikilinks 4 items
- Similar to the idea of TF-IDF

-Choose wikilinks mentioned 4 or more times.
- Helpful reduce noise

```
# Limit to greater than 3 links
links = [t[0] for t in wikilink_counts.items() if t[1] >= 4]
print(len(links))
```

```
41758
```

```
wikilink_counts.get('the new york times')
wikilink_counts.get('new york times')
```

```
2742

996
```

BUILD TRAIN SET

```
index_book[pairs[5000][0]], index_link[pairs[5000][1]]
```

```
('Slaves in the Family', 'category:american biographies')
```

```
index_book[pairs[900][0]], index_link[pairs[900][1]]
```

```
('The Man Who Watched the Trains Go By (novel)',
 'category:belgian novels adapted into films')
```

```
index_book[13337], index_link[31111]
index_book[31899], index_link[65]
index_book[25899], index_link[30465]
```

```
("France's Songs of the Bards of the Tyne - 1850", 'joseph philip robson')

('The Early Stories: 1953–1975', 'the new yorker')

('Marthandavarma (novel)', 'kerala sahitya akademi')
```

```
[((13337, 31111), 85),
 ((31899, 65), 77),
 ((25899, 8850), 61),
 ((1851, 2629), 57),
 ((25899, 30465), 53)]
```

TRAIN / TEST DATA

TRAIN/TEST SET

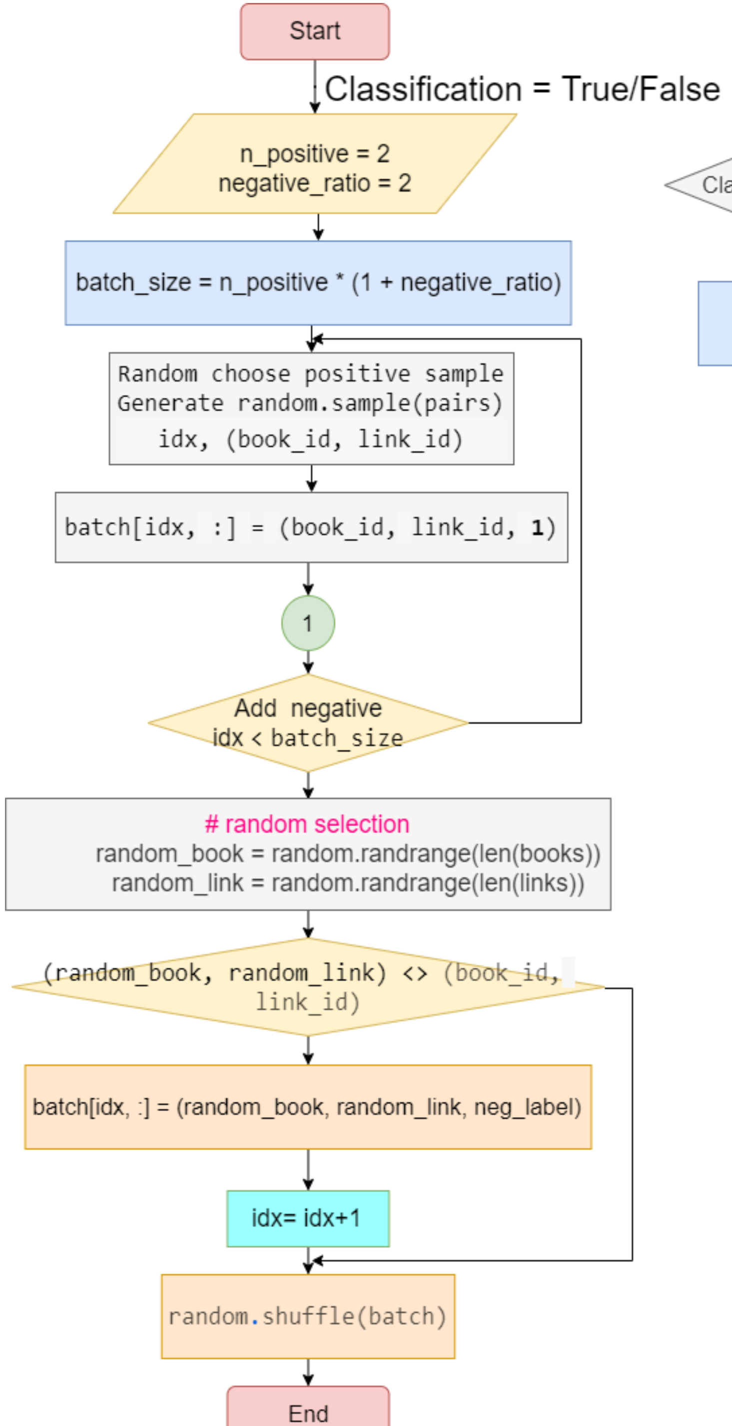
Objective

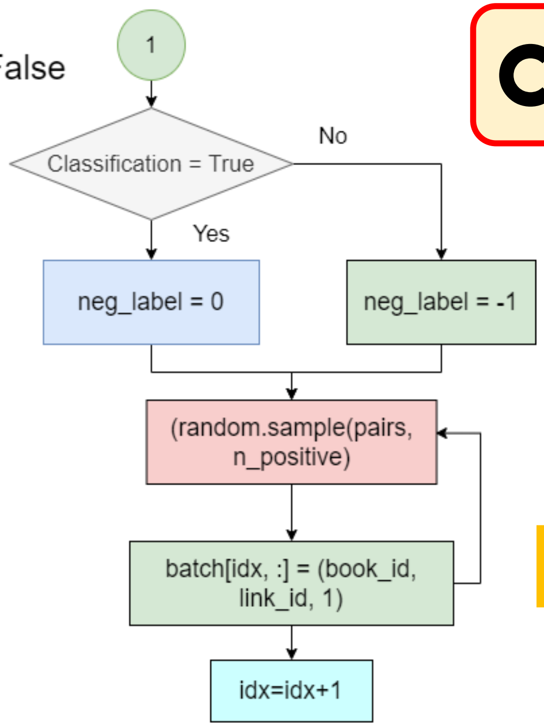
- Our primary objective is not to make the most accurate model, but to generate the best embeddings to train our network .
- Instead of testing on new data, we'll look at the embeddings themselves to see if books that we think are similar have embeddings that are close to each other.
- To separate validation / testing set, then we would be limiting the amount of data that our network can use to train.
- To concerned about overfitting because we do not need our model to generalize to new data and our goal is the embeddings,

GENERATE TRAIN SAMPLE

- To generate positive samples and negative samples to train the neural network.
- The <u>positive samples </u>pick a pair from pairs and assign it <u>a : 1.</u>
- The <u>negative samples </u>pick one random link and one random book, make sure they are <u>not in pairs</u>, and assign them <u>a : -1 or a 0</u>.

CREATE TRAINING PAIRS AND LABEL






```
Book: Deep Six (novel)              Link: president of the united states
Label: 1.0
Book: The Counterfeit Man           Link: gerald gardner (wiccan)
Label: -1.0
Book: Soul Music (novel)            Link: peter crowther
Label: -1.0
Book: The Soul of the Robot         Link: category:house of night series
Label: -1.0
Book: Des Imagistes                 Link: august strindberg
Label: -1.0
Book: Bag of Bones                  Link: category:novels by stephen king
Label: 1.0
```

Remark :

- Over 770,000 positive examples.
- The negative example will random in sample and the results pair is not in pairs.

Neural Network Embedding Model

NEURAL NETWORK EMBEDDING MODEL

Regression Model

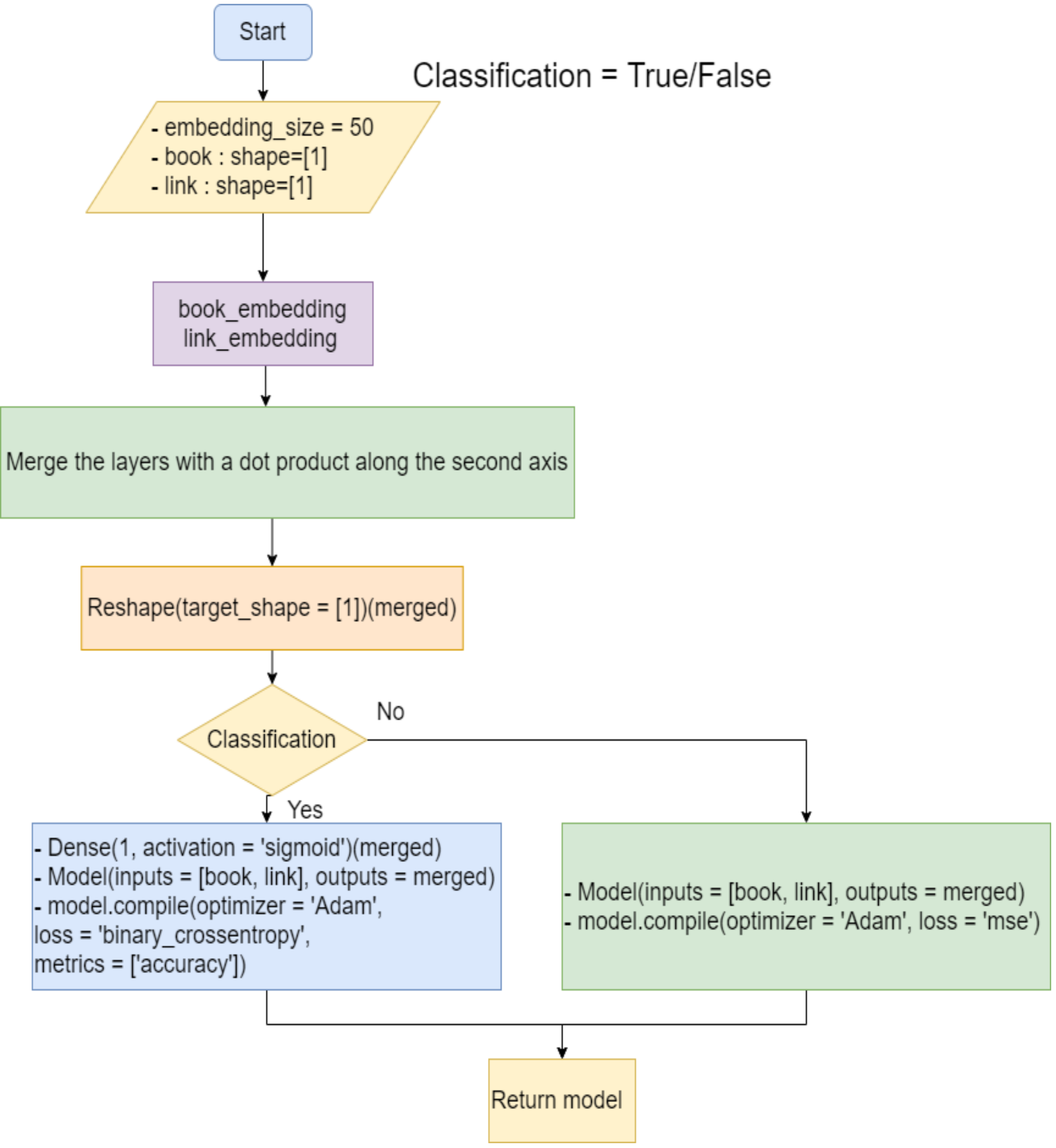
- Our labels are either -1 or 1, using regression model (make mean squared error minimize the distance between the prediction and the output)
- Using the dot product with normalization means dot layer is finding the cosine similarity between the embedding for the book and the link.

NEURAL NETWORK EMBEDDING MODEL

For classification

- Dense layer with a sigmoid activation to squash the outputs between 0 and 1,
- The loss function for classification is binary crossentropy
  - To measures the error of the neural network predictions in a binary classification problem
  - To measure of the similarity between two distributions.
- To calculating the gradients through backpropagation - is Adam in both cases(Adam is a modification to Stochastic Gradient Descent) or updating the model parameters.

BOOK EMBEDDING MODEL




```
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
book (InputLayer)               (None, 1)            0
__________________________________________________________________________________________________
link (InputLayer)               (None, 1)            0
__________________________________________________________________________________________________
book_embedding (Embedding)      (None, 1, 50)        1851000     book[0][0]
__________________________________________________________________________________________________
link_embedding (Embedding)      (None, 1, 50)        2087900     link[0][0]
__________________________________________________________________________________________________
dot_product (Dot)               (None, 1, 1)         0           book_embedding[0][0]
                                                                 link_embedding[0][0]
__________________________________________________________________________________________________
reshape_1 (Reshape)             (None, 1)            0           dot_product[0][0]
==================================================================================================
Total params: 3,938,900
Trainable params: 3,938,900
Non-trainable params: 0
__________________________________________________________________________________________________
```

Remark : There are nearly 4.0 million weights (parameters) that need to be learned by the neural network

TRAIN MODEL

```
n_positive = 1024

gen = generate_batch(pairs, n_positive, negative_ratio = 2)

# Train
h = model.fit_generator(gen, epochs = 15,
                        steps_per_epoch = len(pairs) // n_positive,
                        verbose = 2)
```

```
Epoch 1/15
 - 16s - loss: 0.9625
Epoch 2/15
 - 15s - loss: 0.7629
Epoch 3/15
 - 15s - loss: 0.5419
Epoch 4/15
 - 15s - loss: 0.5022
Epoch 5/15
 - 15s - loss: 0.4791
Epoch 6/15
 - 15s - loss: 0.4745
Epoch 7/15
 - 15s - loss: 0.4633
Epoch 8/15
 - 15s - loss: 0.4680
Epoch 9/15
 - 15s - loss: 0.4640
Epoch 10/15
 - 15s - loss: 0.4590
Epoch 11/15
 - 15s - loss: 0.4511
Epoch 12/15
 - 15s - loss: 0.4478
Epoch 13/15
 - 15s - loss: 0.4489
Epoch 14/15
 - 15s - loss: 0.4477
Epoch 15/15
 - 15s - loss: 0.4506
```

Save Model : first_attempt.h5

EXTRACT EMBEDDING ANALYZE

```
# Extract embeddings
book_layer = model.get_layer('book_embedding')
book_weights = book_layer.get_weights()[0]
book_weights.shape
```

```
(37020, 50)
```

Calculation: Book frequency 37020
50-dimensional vector.

```
book_weights = book_weights / np.linalg.norm(book_weights, axis = 1).reshape((-1, 1))
book_weights[0][:10]
np.sum(np.square(book_weights[0]))
```

```
array([ 0.10427548, -0.23591727,  0.14965919, -0.07087466,  0.09660177,
       -0.20169103,  0.05245946, -0.08891259, -0.12968971, -0.03209771],
      dtype=float32)

1.0
```

To normalize the embeddings
so that the dot product between two embeddings becomes
the cosine similarity

Find Similarity

COSINE SIMILARITY

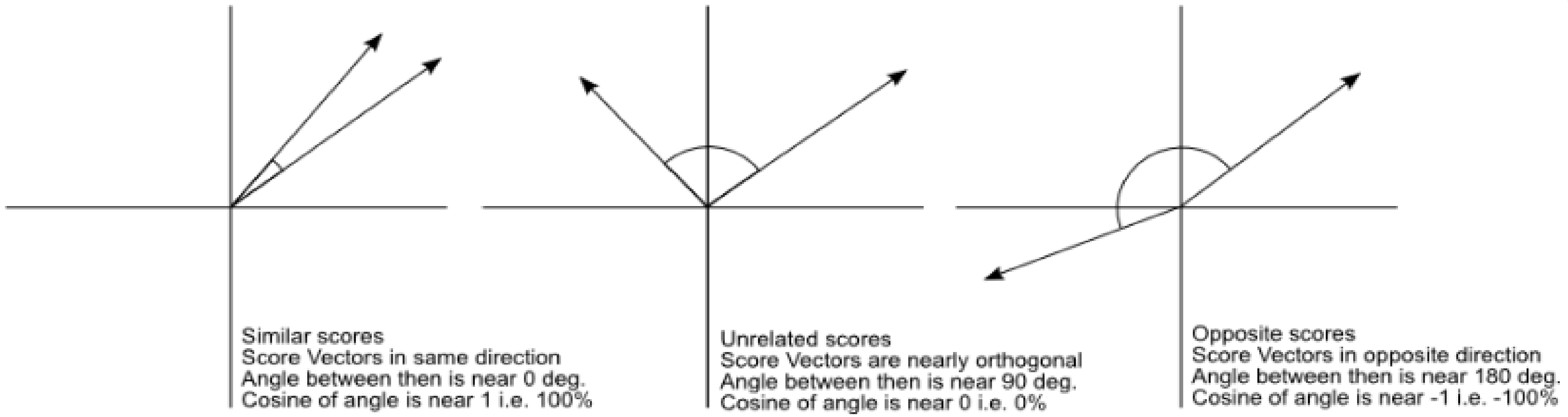
$$\vec{a}\cdot\vec{b} = \sum_{i=1}^{n} a_i b_i = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$$

$$\cos\theta = \frac{\vec{a}\cdot\vec{b}}{\|\vec{a}\|\|\vec{b}\|}$$

$$\|\vec{a}\| = \sqrt{a_1^2 + a_2^2 + a_3^2 + \cdots + a_n^2}$$

$$\|\vec{b}\| = \sqrt{b_1^2 + b_2^2 + b_3^2 + \cdots + b_n^2}$$

Mathematically, if 'a' and 'b' are two vectors, cosine equation gives the angle between the two.



The Cosine Similarity values for different documents, 1 (same direction), 0 (90 deg.), -1 (opposite directions).

From : https://medium.com/acing-ai/what-is-cosine-similarity-matrix-f0819e674ad1

BOOK

BOOKS :WAR AND PEACE

Classification = False (RG)

Books closest to War and Peace.

Book: War and Peace	Similarity: 1.0
Book: Anna Karenina	Similarity: 0.92
Book: The Master and Margarita	Similarity: 0.92
Book: Buddenbrooks	Similarity: 0.9
Book: Crime and Punishment	Similarity: 0.89
Book: Demons (Dostoevsky novel)	Similarity: 0.89
Book: Candide	Similarity: 0.88
Book: The Hunchback of Notre-Dame	Similarity: 0.87
Book: The Magic Mountain	Similarity: 0.87
Book: Don Quixote	Similarity: 0.87

Classification =True (DL)

Books closest to War and Peace.

Book: War and Peace	Similarity: 1.0
Book: Anna Karenina	Similarity: 0.83
Book: Doctor Zhivago (novel)	Similarity: 0.74
Book: Dead Souls	Similarity: 0.74
Book: Eugene Onegin	Similarity: 0.71
Book: The Master and Margarita	Similarity: 0.7
Book: Candide	Similarity: 0.7
Book: Demons (Dostoevsky novel)	Similarity: 0.69
Book: The Bronze Horseman (poem)	Similarity: 0.68
Book: The Brothers Karamazov	Similarity: 0.67

From : http://blog.christianperone.com/2013/09/machine-learning-cosine-similarity-for-vector-space-models-part-iii/

BOOKS : ARTIFICIAL INTELLIGENCE(AI)

Classification = False (RG)

Books closest to Artificial Intelligence: A Modern Approach.

Book: Artificial Intelligence: A Modern Approach Similarity: 1.0
Book: Essentials of Programming Languages Similarity: 0.95
Book: Computer Graphics: Principles and Practice Similarity: 0.95
Book: TCP/IP Illustrated Similarity: 0.94
Book: Structure and Interpretation of Computer Programs Similarity: 0.94
Book: Compilers: Principles, Techniques, and Tools Similarity: 0.93
Book: Lions' Commentary on UNIX 6th Edition, with Source Code Similarity: 0.93
Book: The Linux Programming Interface Similarity: 0.93
Book: Algorithms + Data Structures = Programs Similarity: 0.92
Book: Lisp in Small Pieces Similarity: 0.92

Classification =True (DL)

Books closest to Artificial Intelligence: A Modern Approach.

Book: Artificial Intelligence: A Modern Approach Similarity: 1.0
Book: Structure and Interpretation of Computer Programs Similarity: 0.8
Book: The Linux Programming Interface Similarity: 0.8
Book: Code: The Hidden Language of Computer Hardware and Software Similarity: 0.79
Book: The Practice of Programming Similarity: 0.79
Book: Computer Graphics: Principles and Practice Similarity: 0.79
Book: Algorithms Unlocked Similarity: 0.78
Book: Operating Systems: Design and Implementation Similarity: 0.78
Book: Algorithms + Data Structures = Programs Similarity: 0.77
Book: The Cult of Mac Similarity: 0.77

BOOKS :WEAPONS OF MATH DESTRUCTION

Classification = False (RG)

Books closest to Weapons of Math Destruction.

Book: Weapons of Math Destruction	Similarity: 1.0
Book: The Alchemy of Race and Rights	Similarity: 0.94
Book: Affirmative Action Around the World	Similarity: 0.93
Book: Conscience and Its Enemies	Similarity: 0.93
Book: American Nietzsche	Similarity: 0.92
Book: The Sexual Paradox	Similarity: 0.92
Book: Huck's Raft	Similarity: 0.92
Book: The Vision of the Anointed	Similarity: 0.91
Book: Intelligence and How to Get It	Similarity: 0.91
Book: Linked: The New Science of Networks	Similarity: 0.91

Classification =True (DL)

Books closest to Weapons of Math Destruction.

Book: Weapons of Math Destruction	Similarity: 1.0
Book: O Strange New World	Similarity: 0.77
Book: The Soul of a New Machine	Similarity: 0.75
Book: On Immunity: An Inoculation	Similarity: 0.74
Book: Annals of the Former World	Similarity: 0.73
Book: Legacy of Ashes (book)	Similarity: 0.72
Book: Ordinary Light	Similarity: 0.72
Book: The Shallows (book)	Similarity: 0.72
Book: How to Be Black	Similarity: 0.71
Book: Race: The Reality of Human Difference	Similarity: 0.71

BOOKS :THE FELLOWSHIP OF THE RING

Classification = False (RG)

Books closest to The Fellowship of the Ring.

Book: The Fellowship of the Ring	Similarity: 1.0
Book: The Return of the King	Similarity: 0.96
Book: The Two Towers	Similarity: 0.91
Book: Beren and Lúthien	Similarity: 0.9
Book: The Silmarillion	Similarity: 0.9
Book: Bored of the Rings	Similarity: 0.88
Book: The History of The Lord of the Rings	Similarity: 0.87
Book: The Book of Lost Tales	Similarity: 0.87
Book: The Lays of Beleriand	Similarity: 0.84
Book: Morgoth's Ring	Similarity: 0.84

Classification =True (DL)

Books closest to The Fellowship of the Ring.

Book: The Fellowship of the Ring	Similarity: 1.0
Book: The Two Towers	Similarity: 0.9
Book: The Return of the King	Similarity: 0.89
Book: The Silmarillion	Similarity: 0.84
Book: The Children of Húrin	Similarity: 0.83
Book: The History of The Lord of the Rings	Similarity: 0.81
Book: The Book of Lost Tales	Similarity: 0.8
Book: The Hobbit	Similarity: 0.8
Book: The War of the Jewels	Similarity: 0.79
Book: Tales from the Perilous Realm	Similarity: 0.78


Symptoms
Anyone who has these symptoms must stay at home until the symptoms
for at least seven days. Everyone else in the
at least 14 days after the first person's
do not have symptoms,
during that time, that individual must
seven days from when they developed
have passed and provided symptoms have
no longer need to isolate.
Do not go to a GP surgery, pharmacy or hospital.
NHS online services. Only call 111 if you are not able to get
you have been instructed to call, or your symptoms worsen.
DAY
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
PERSON A
PERSON B
PERSON C
PERSON D
Develops symptoms, triggering 7-day isolation for herself and 14 days for her household
Isolation ends if symptoms have stopped
Develops symptoms and starts 7-day count
Isolation ends if symptoms have stopped
PAGES

PAGES : WASHINGTON POST

Classification = False (RG)

Pages closest to the washington post.

Page: the washington post	Similarity: 1.0
Page: los angeles times	Similarity: 0.98
Page: san francisco chronicle	Similarity: 0.98
Page: washington post	Similarity: 0.98
Page: the new york times	Similarity: 0.98
Page: npr	Similarity: 0.97
Page: new york times	Similarity: 0.97
Page: memoir	Similarity: 0.96
Page: simon & schuster	Similarity: 0.95
Page: the new yorker	Similarity: 0.94

Classification =True (DL)

Pages closest to the washington post.

Page: the washington post	Similarity: 1.0
Page: los angeles times	Similarity: 0.95
Page: the new york times	Similarity: 0.93
Page: time (magazine)	Similarity: 0.92
Page: washington post	Similarity: 0.91
Page: new york times	Similarity: 0.9
Page: the wall street journal	Similarity: 0.9
Page: the guardian	Similarity: 0.9
Page: san francisco chronicle	Similarity: 0.89
Page: the new yorker	Similarity: 0.88

PAGES CLOSET TO SCIENCE FICTION

Classification = False (RG)

Pages closest to science fiction.

Page	Similarity
Page: science fiction	Similarity: 1.0
Page: category:american science fiction novels	Similarity: 0.98
Page: tor books	Similarity: 0.94
Page: ballantine books	Similarity: 0.92
Page: category:ace books books	Similarity: 0.92
Page: ace books	Similarity: 0.91
Page: category:ballantine books books	Similarity: 0.9
Page: category:doubleday (publisher) books	Similarity: 0.9
Page: victor gollancz ltd	Similarity: 0.9
Page: anthology	Similarity: 0.9

Classification =True (DL)

Pages closest to science fiction.

Page	Similarity
Page: science fiction	Similarity: 1.0
Page: category:american science fiction novels	Similarity: 0.94
Page: ballantine books	Similarity: 0.83
Page: del rey books	Similarity: 0.8
Page: category:time travel novels	Similarity: 0.78
Page: ace books	Similarity: 0.77
Page: category:ace books books	Similarity: 0.77
Page: bantam books	Similarity: 0.76
Page: category:dystopian novels	Similarity: 0.76
Page: short story	Similarity: 0.76

PAGES : NEW YORK CITY

Classification = False (RG)

Pages closest to new york city.

Page: new york city	Similarity: 1.0
Page: the new york times	Similarity: 0.96
Page: alfred a. knopf	Similarity: 0.96
Page: category:random house books	Similarity: 0.95
Page: category:alfred a. knopf books	Similarity: 0.95
Page: random house	Similarity: 0.94
Page: simon & schuster	Similarity: 0.94
Page: new york times	Similarity: 0.94
Page: little, brown and company	Similarity: 0.93
Page: los angeles times	Similarity: 0.92

Classification =True (DL)

Pages closest to new york city.

Page: new york city	Similarity: 1.0
Page: the new york times	Similarity: 0.91
Page: random house	Similarity: 0.89
Page: los angeles times	Similarity: 0.87
Page: simon & schuster	Similarity: 0.87
Page: new york times	Similarity: 0.87
Page: time (magazine)	Similarity: 0.86
Page: united states	Similarity: 0.85
Page: world war ii	Similarity: 0.84
Page: alfred a. knopf	Similarity: 0.84

Data visualization

TSNE AND UMAP

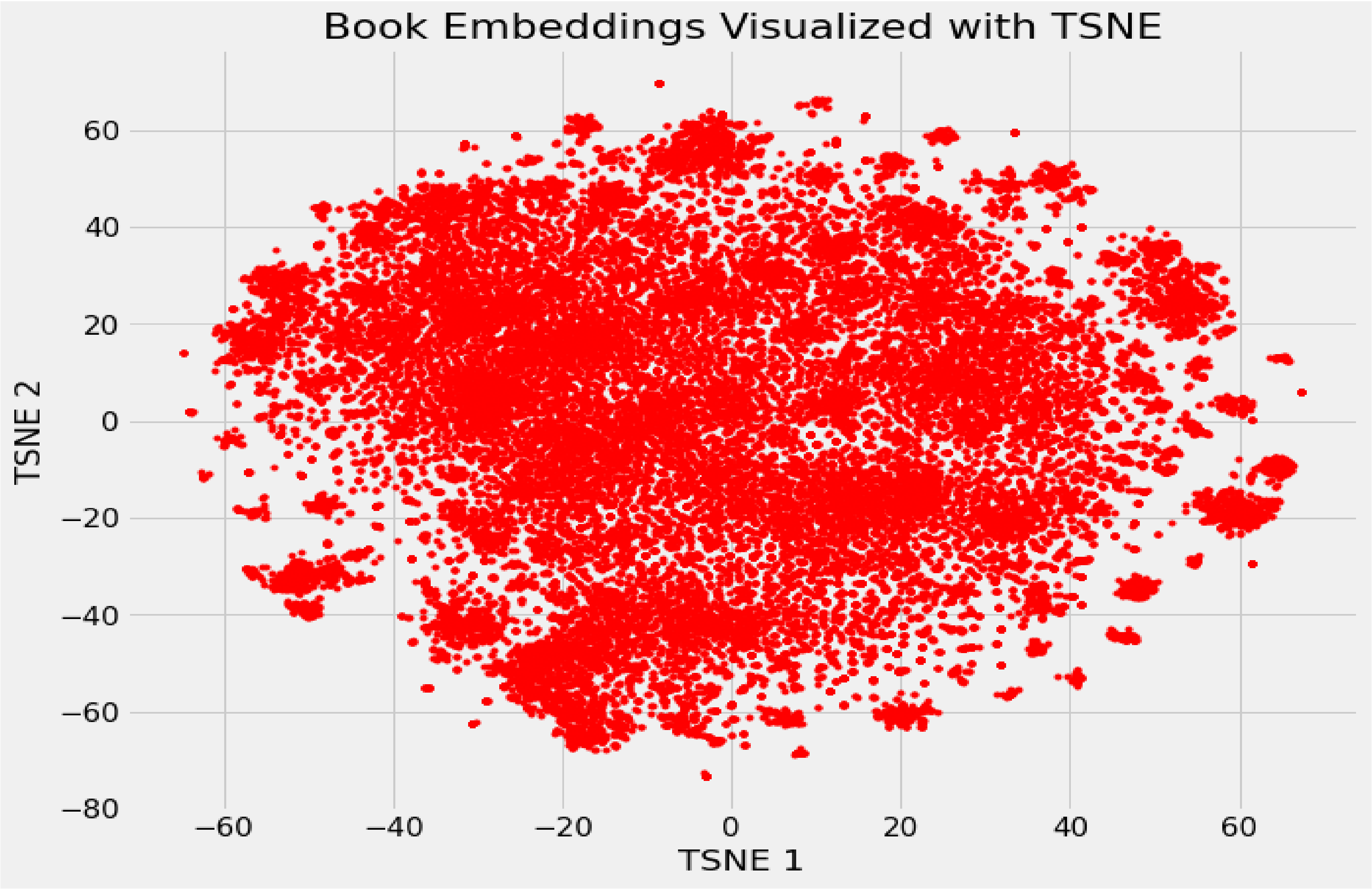
In [59]:
```
from sklearn.manifold import TSNE
from umap import UMAP
```

In [60]:
```
def reduce_dim(weights, components = 3, method = 'tsne'):
    """Reduce dimensions of embeddings"""
    if method == 'tsne':
        return TSNE(components, metric = 'cosine').fit_transform(weights)
    elif method == 'umap':
        # Might want to try different parameters for UMAP
        return UMAP(n_components=components, metric = 'cosine',
                    init = 'random', n_neighbors = 5).fit_transform(weights)
```

In [61]:
```
book_r = reduce_dim(book_weights_class, components = 2, method = 'tsne')
book_r.shape
```

Out[61]:
```
(37020, 2)
```

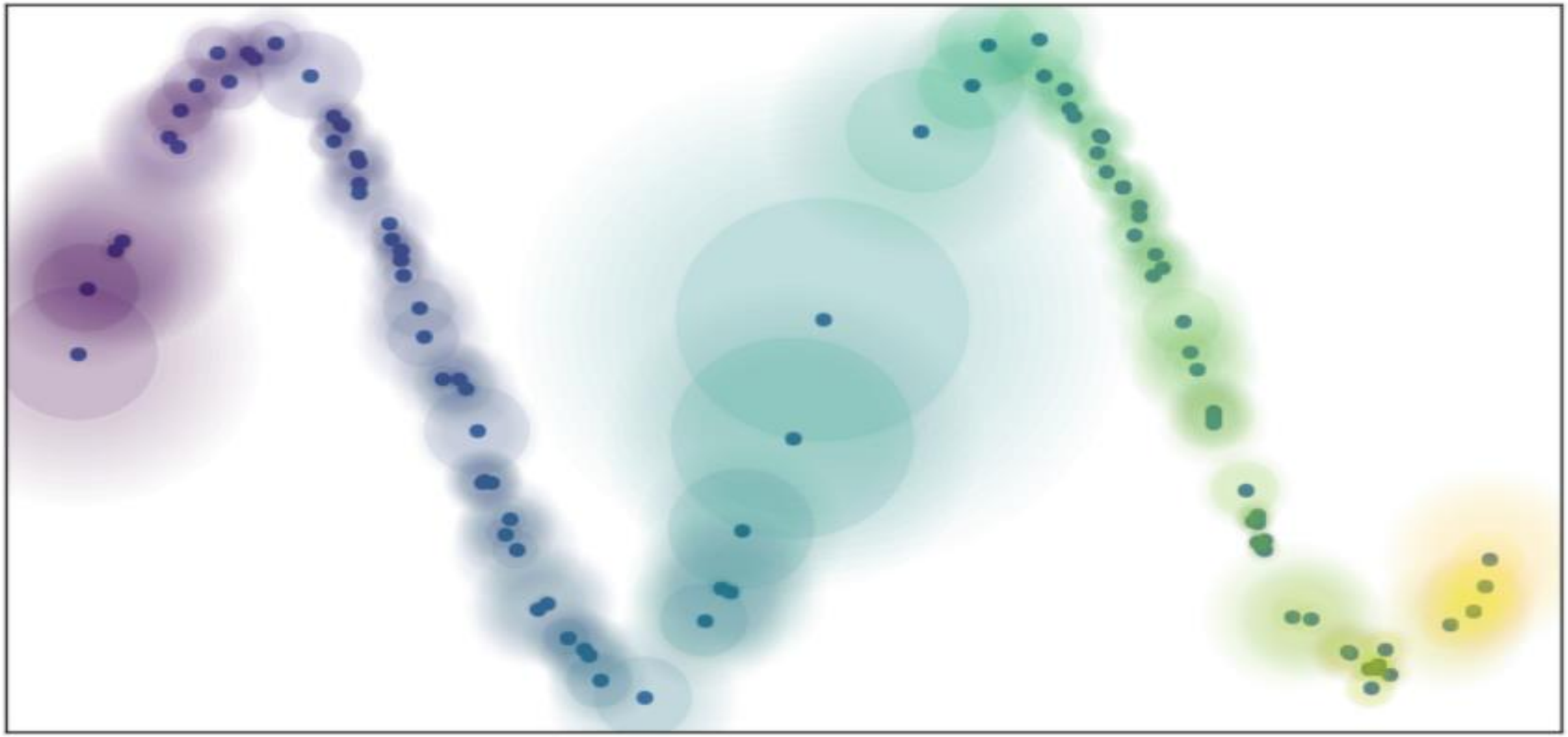
TSNE: t-Stochastic Distributed Neighbors Embedding



We've now taken the initial 37,000 dimension book vector and reduced it to just 2 dimensions.

UMAP

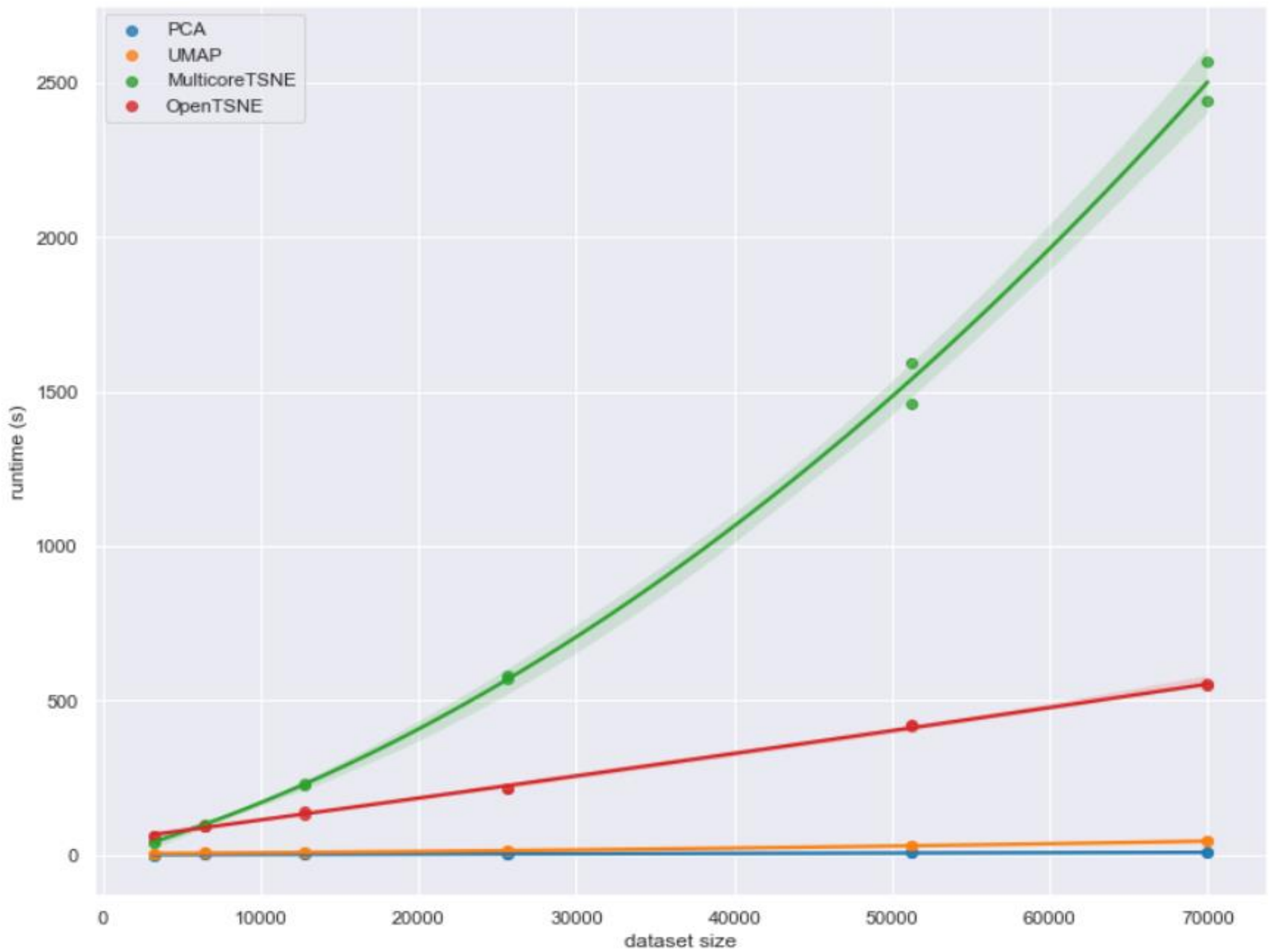
UNIFORM MANIFOLD APPROXIMATION AND PROJECTION



Local connectivity and fuzzy open sets

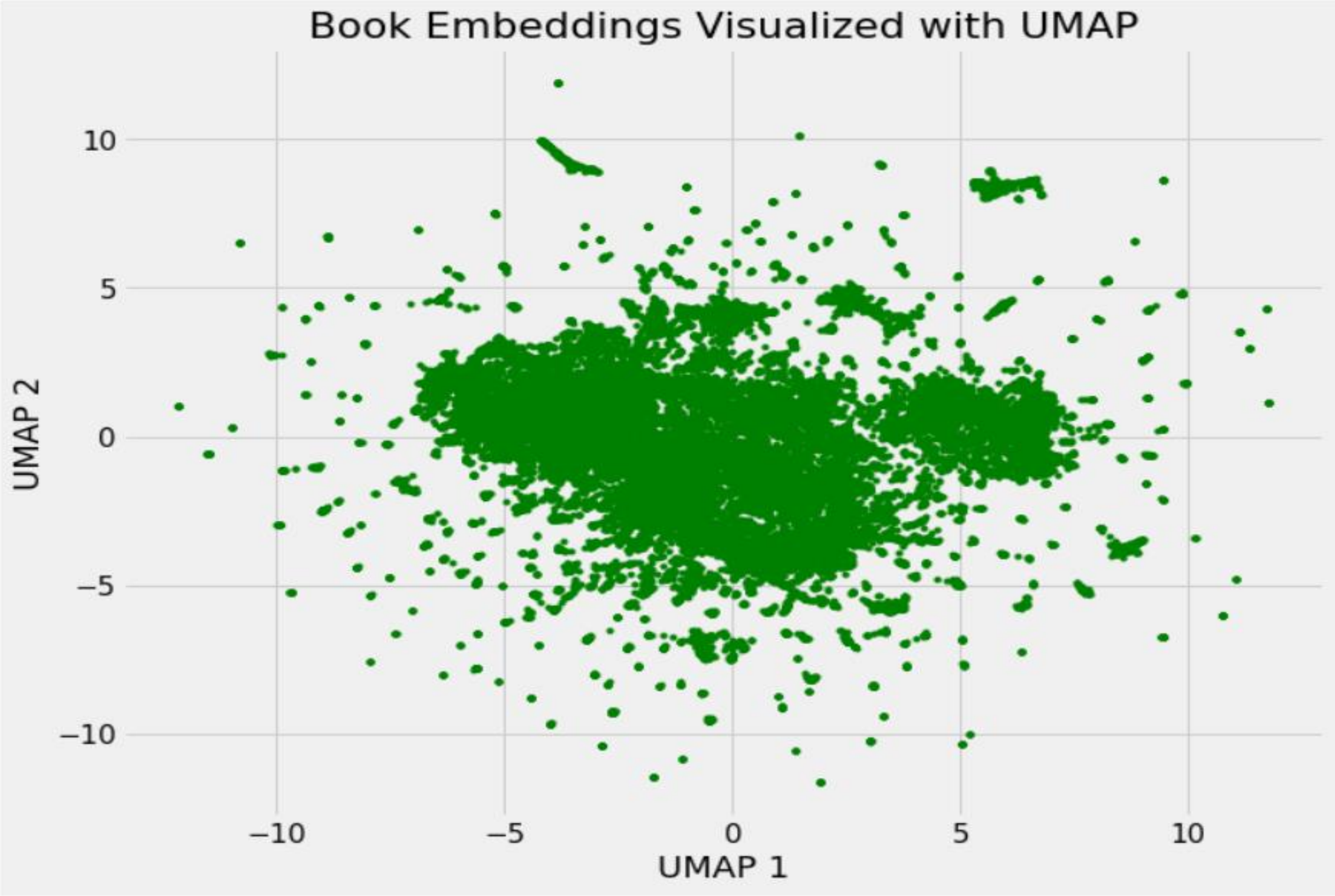
From : https://umap-learn.readthedocs.io/en/latest/how_umap_works.html

PERFORMANCE RUNTIME

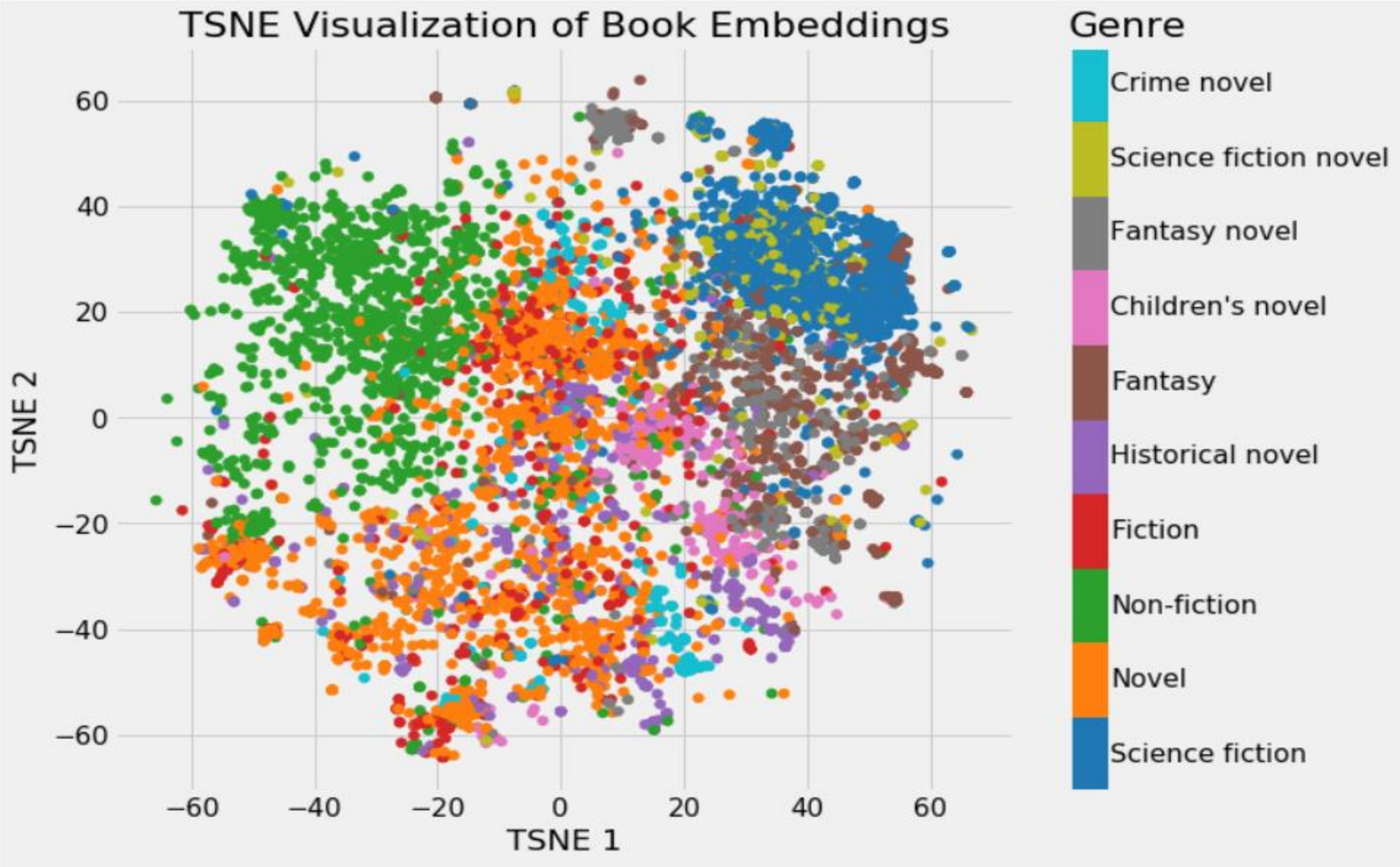


From : https://umap-learn.readthedocs.io/en/latest/performance.html

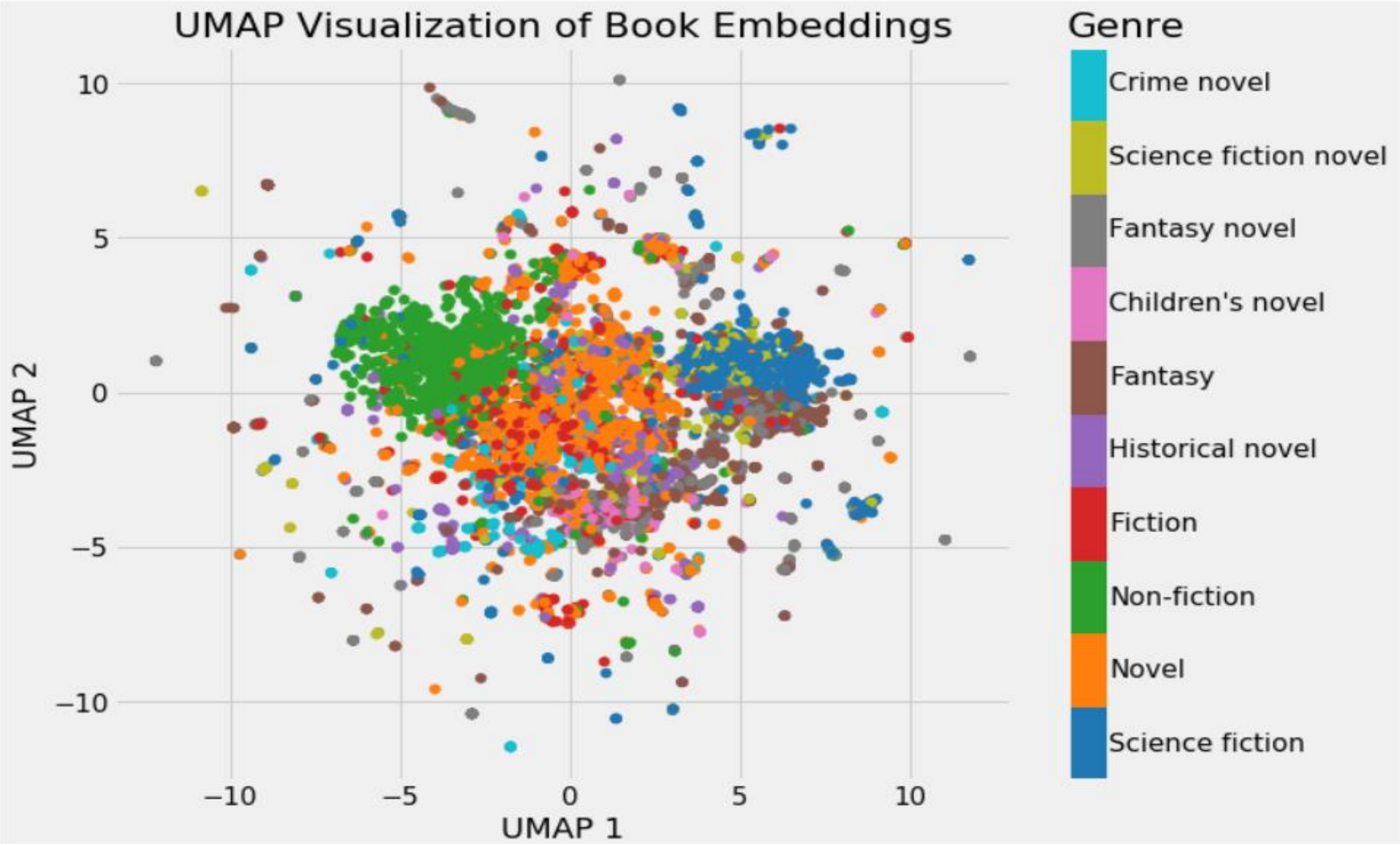
UMAP


Book Embeddings Visualized with UMAP
10
5
0
−5
−10
UMAP 2
−10
−5
0
5
10
UMAP 1

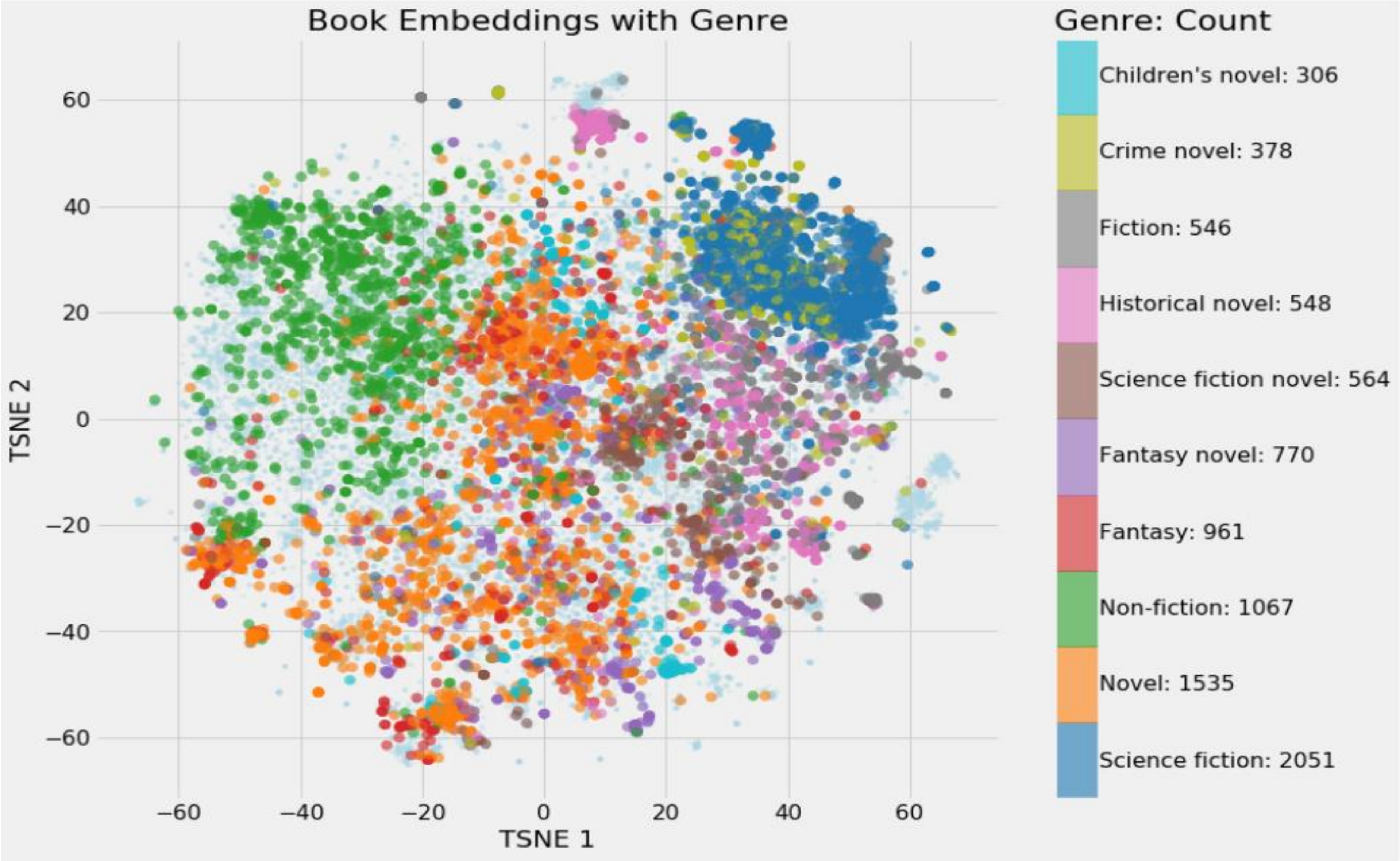
TSNE : BOOK EMBEDDING



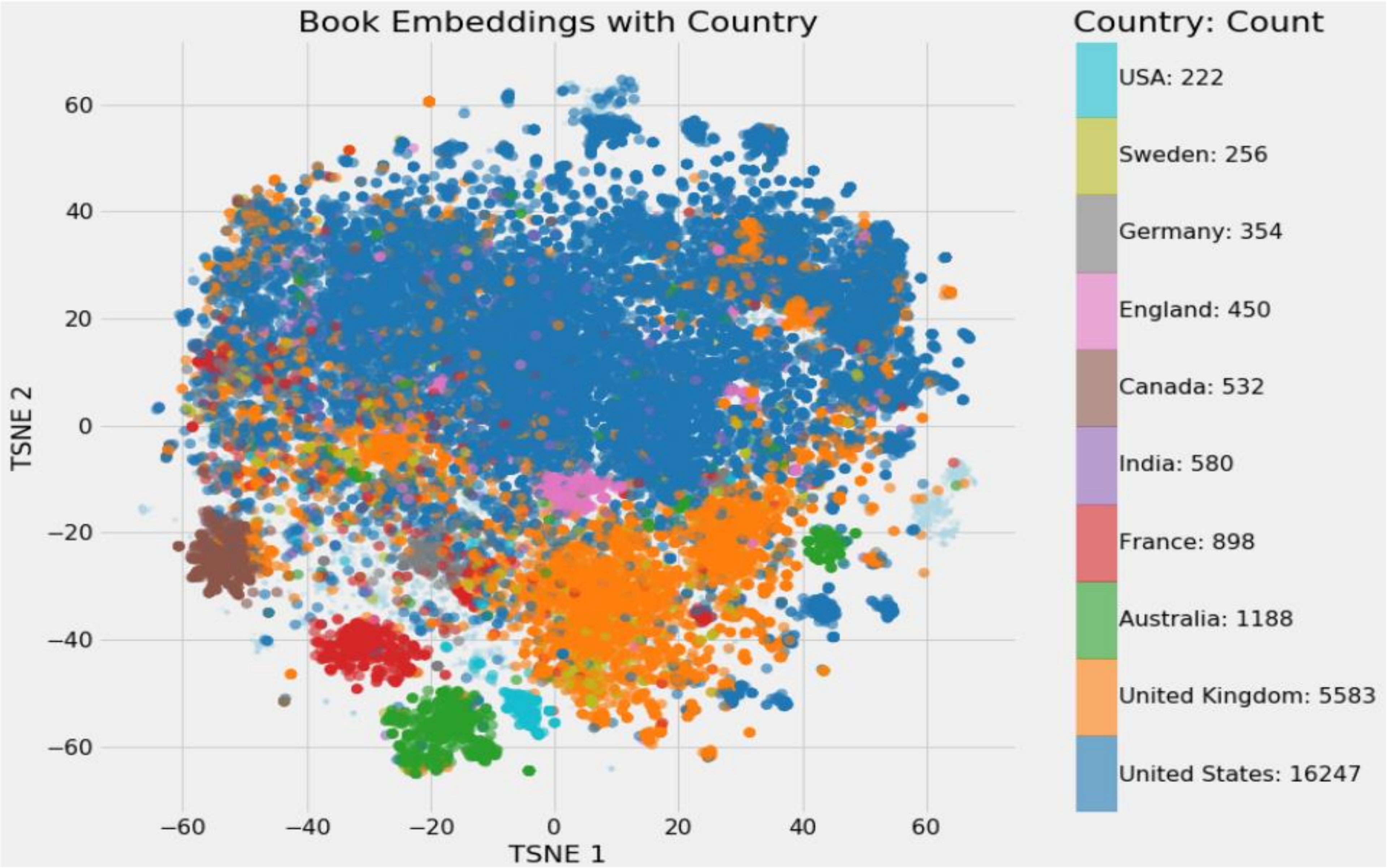
UMAP : BOOK EMBEDDING



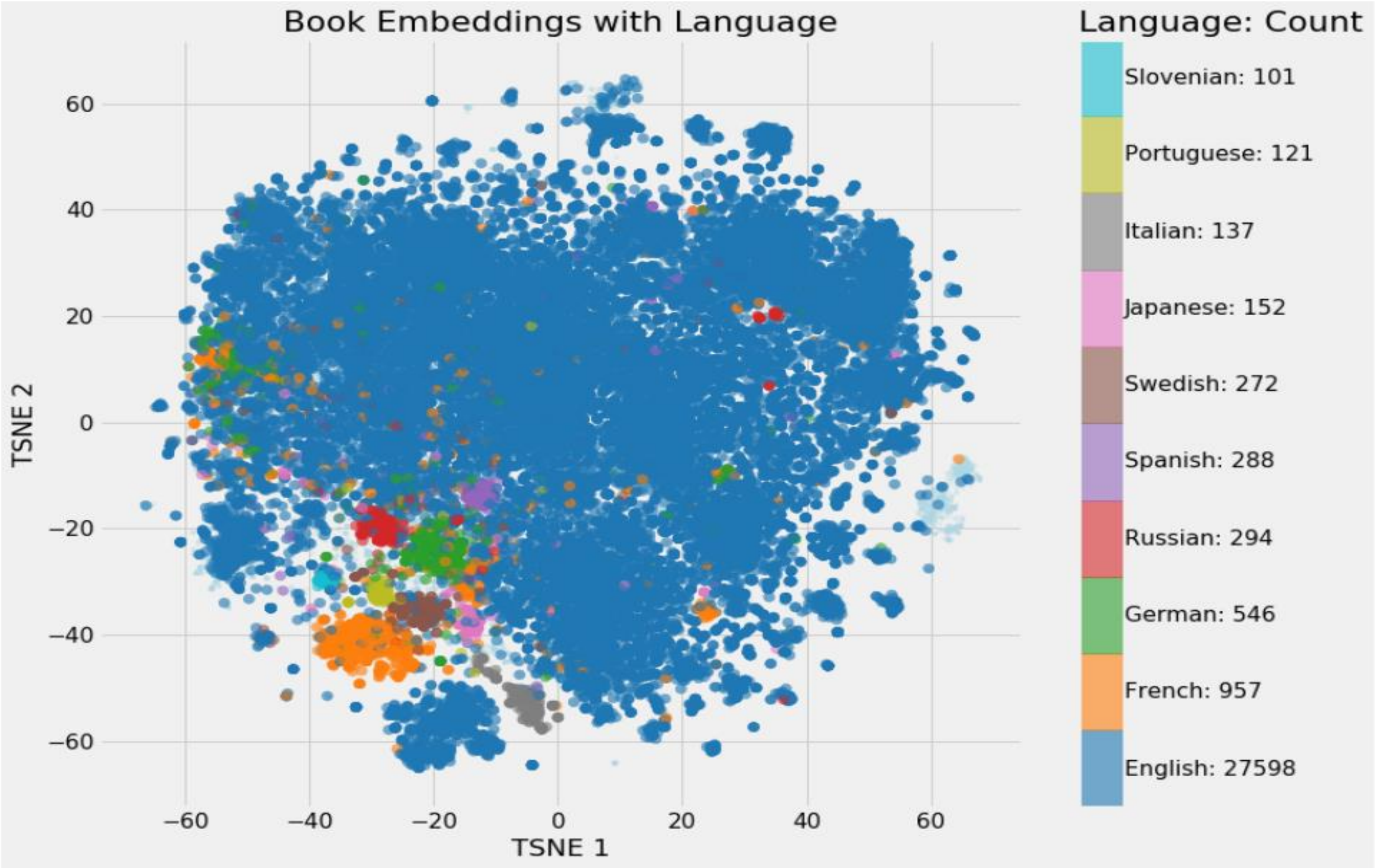
TSNE : BOOK : GENRE ADAPT COLOR



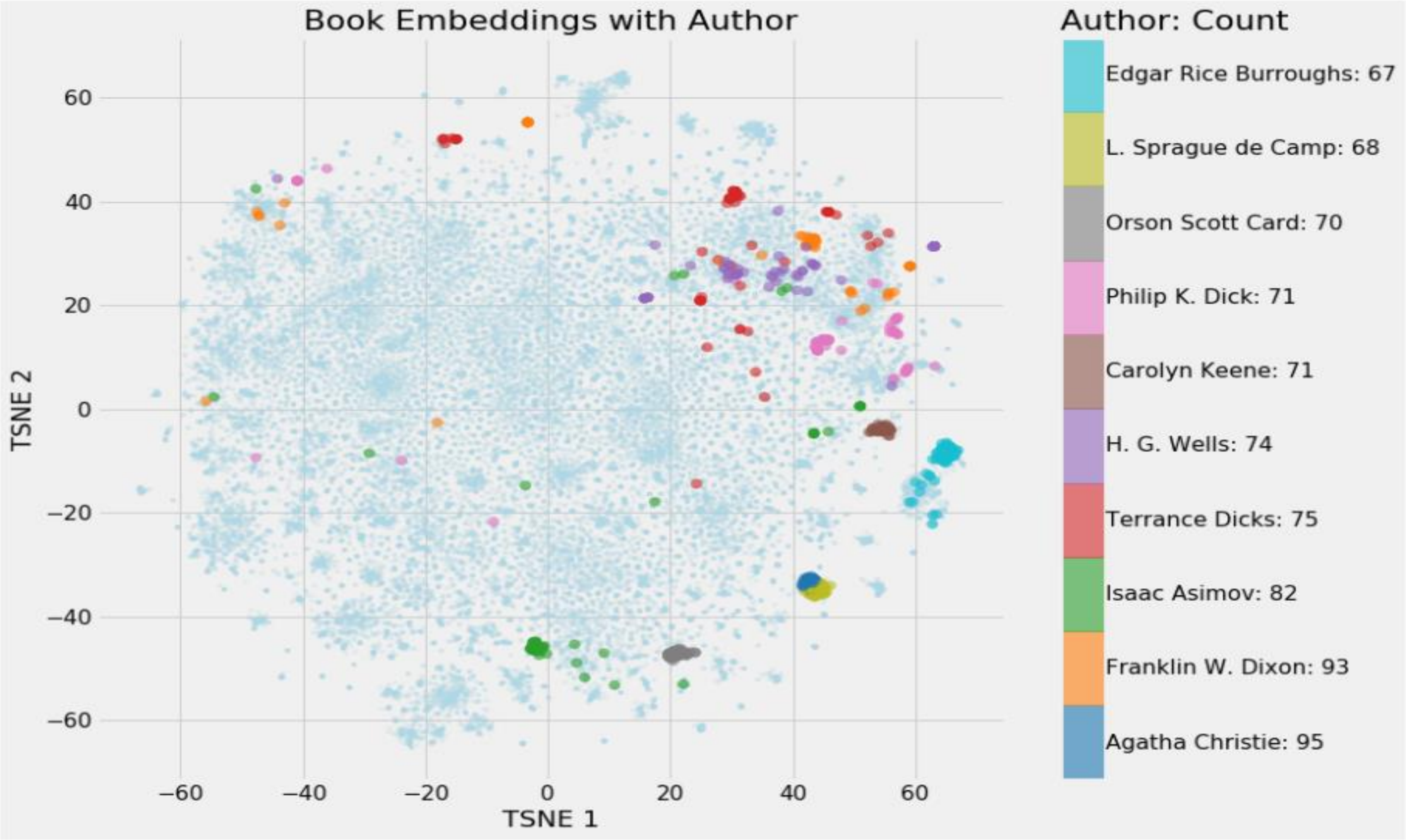
TSNE BOOK : COUNTRY



TSNE BOOK :LANGUAGE



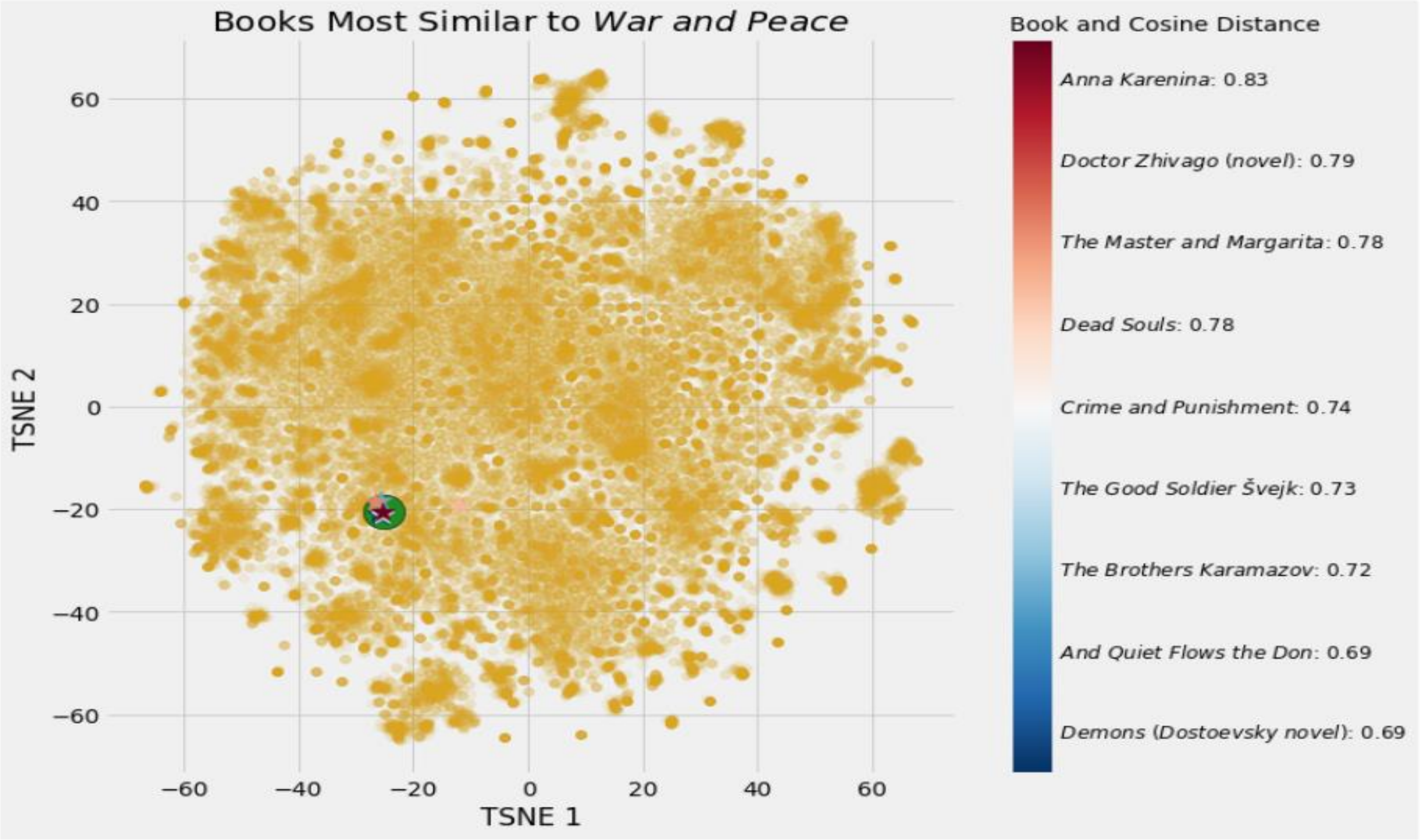
TSNE BOOK : AUTHOR



PLOT BOOK NEAREST NEIGHBORS

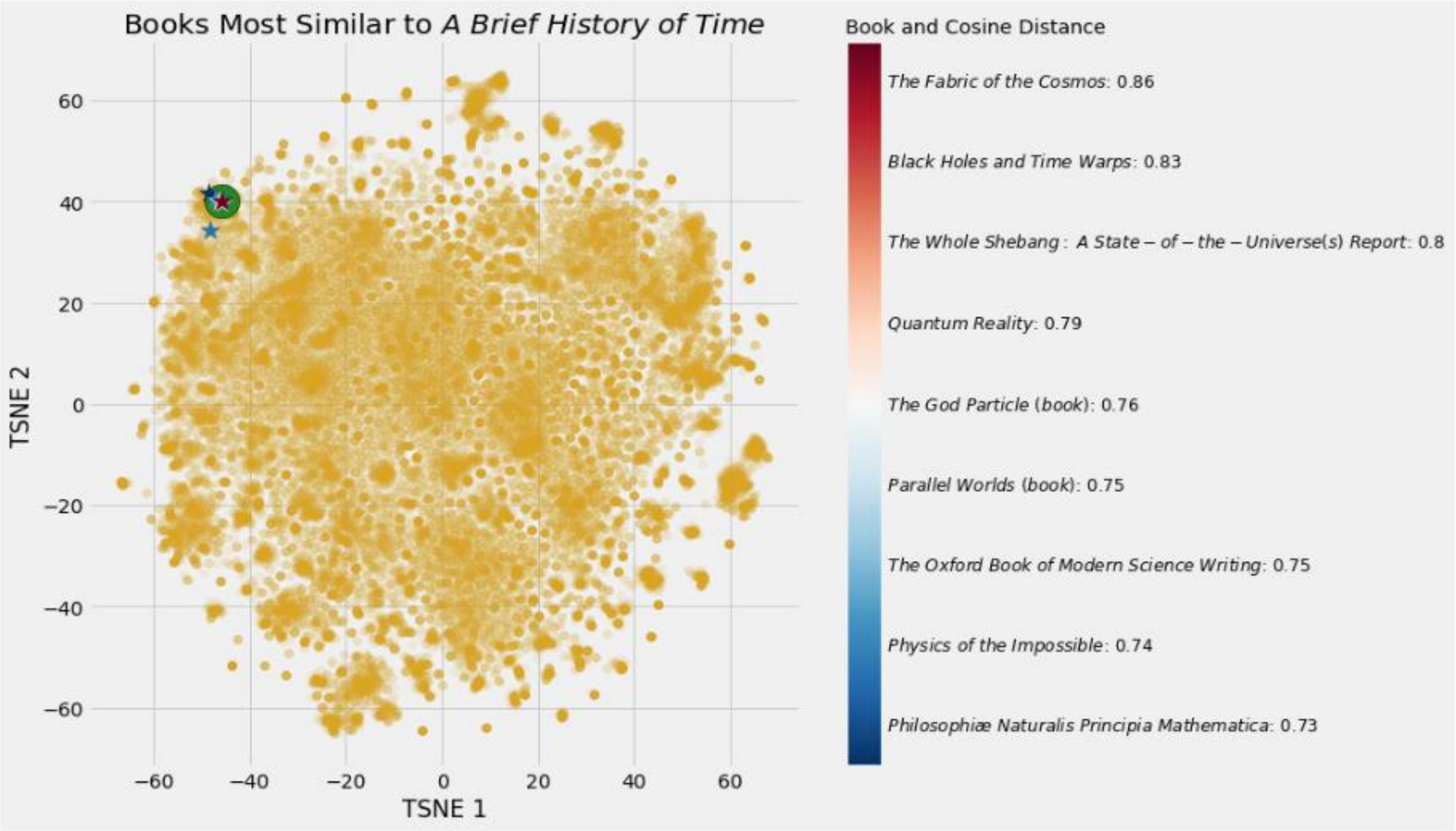
TSNE BOOK : WAR AND PEACE 9 ITEMS

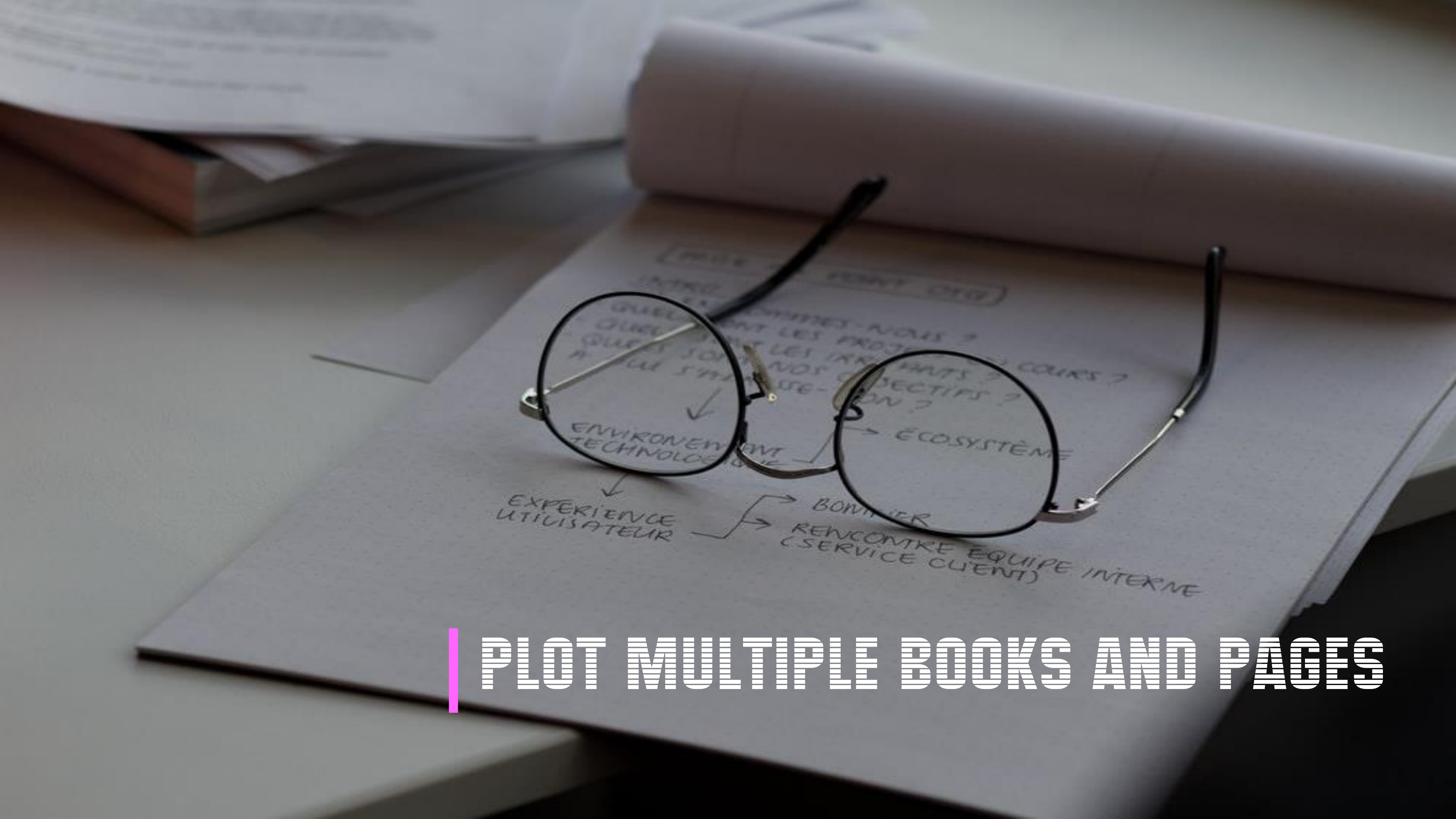
We can see that even though these are the closest books in the 50-dimensional embedding space, when we reduce it down to 2 dimensions, the same separations are not preserved.



TSNE BOOKS : BRIEF HISTORY OF TIME 9 ITEMS

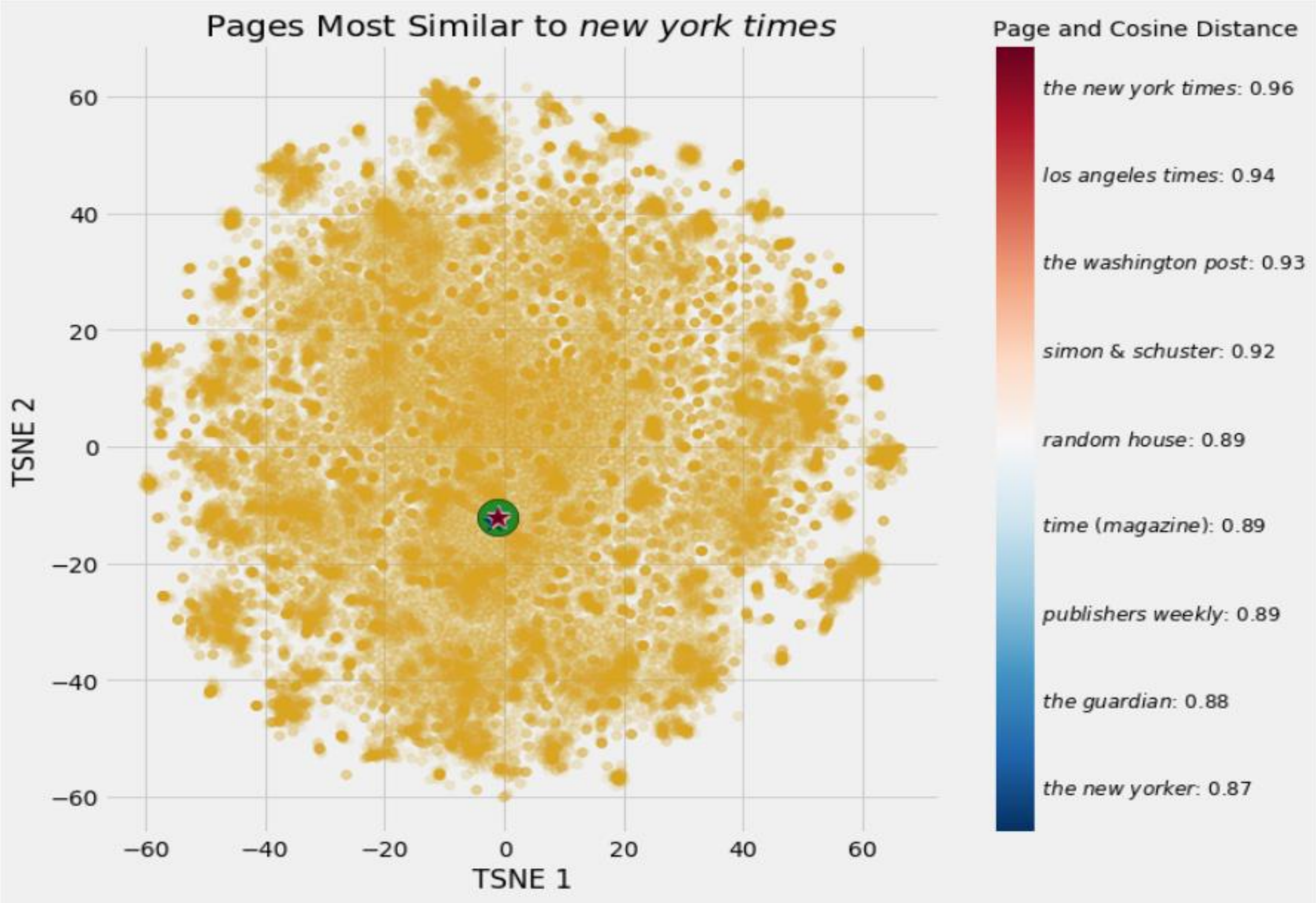
10 most popular categories



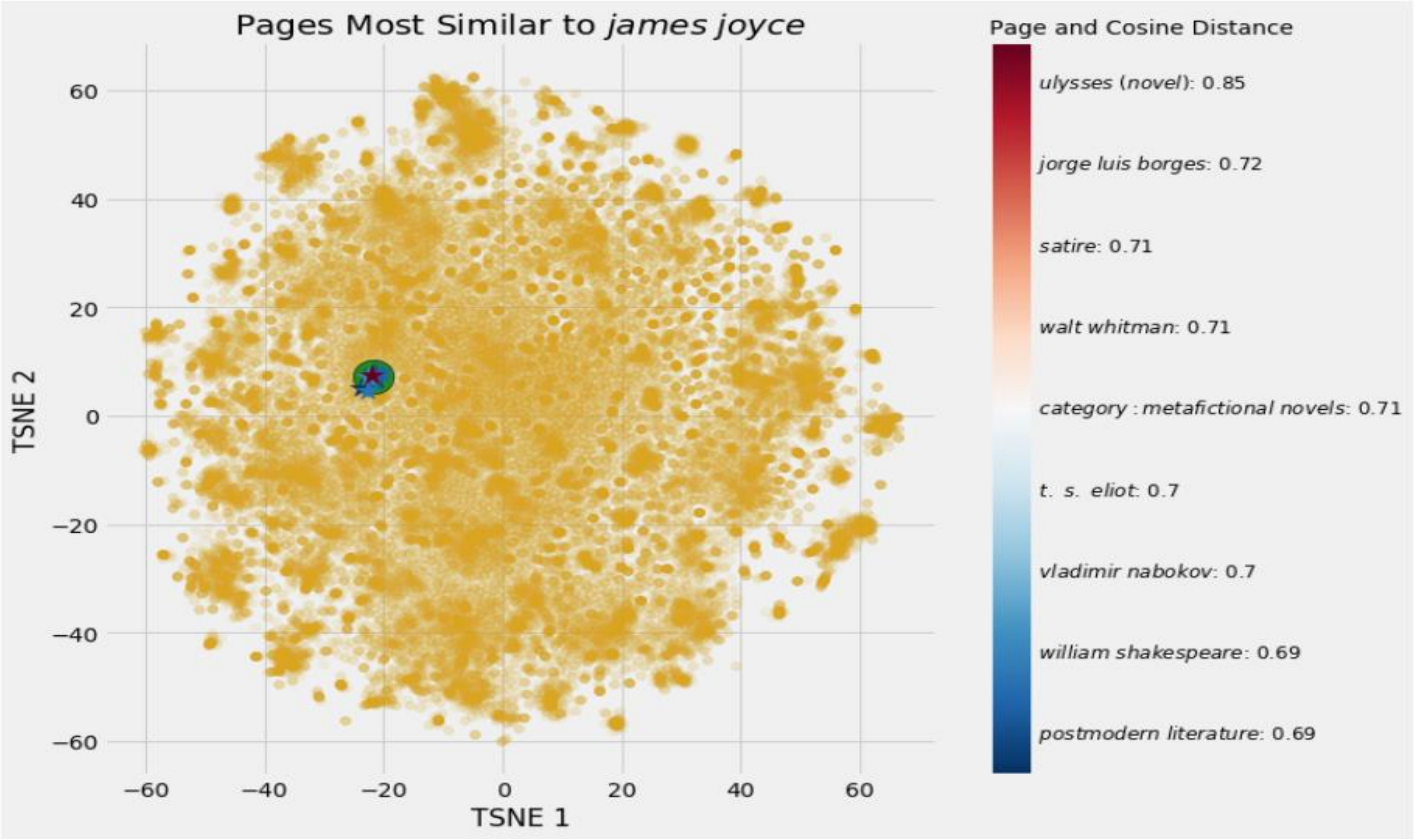


PLOT MULTIPLE BOOKS AND PAGES

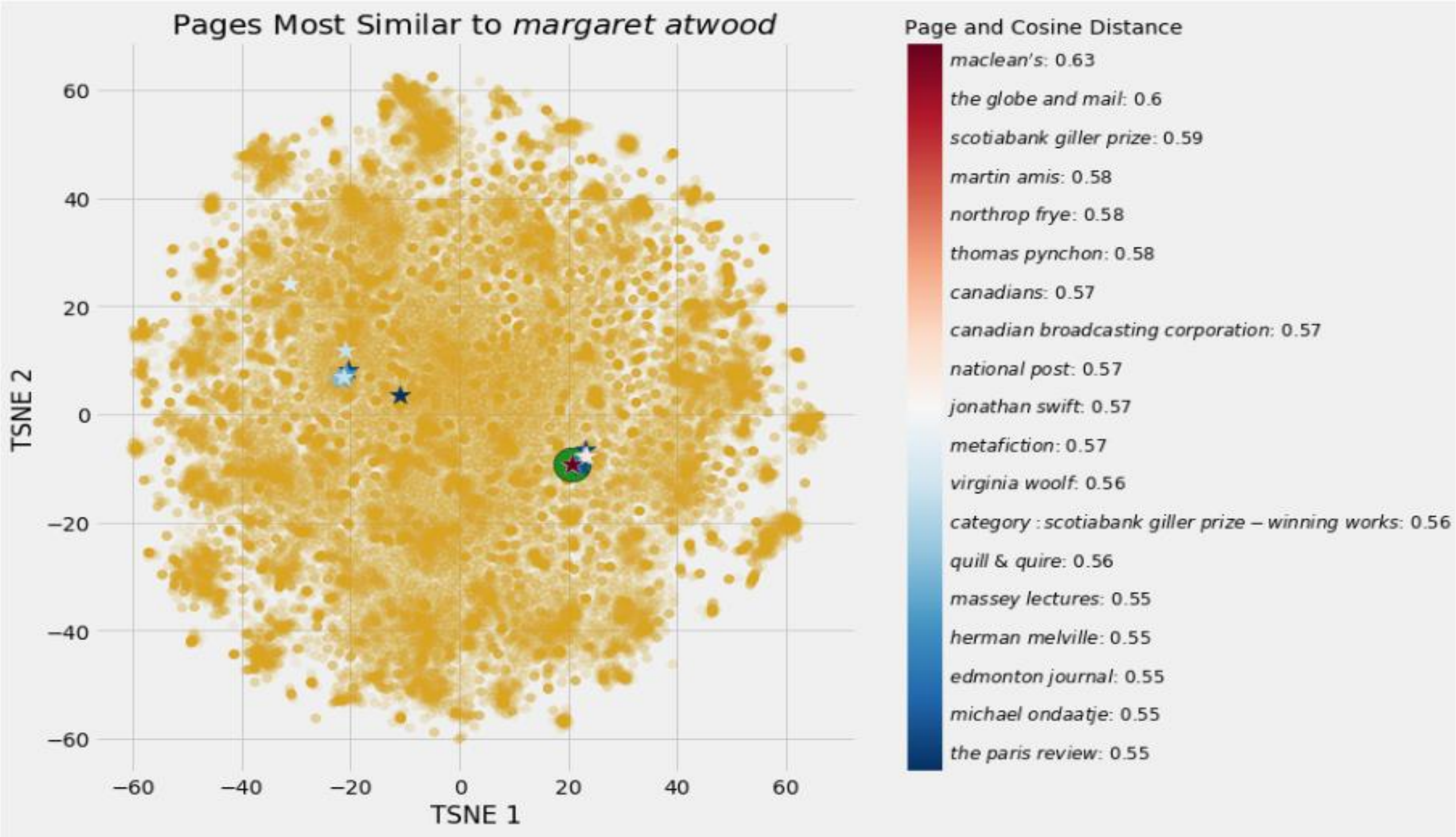
TSNE PAGES : NEW YORK TIMES 9 ITEMS



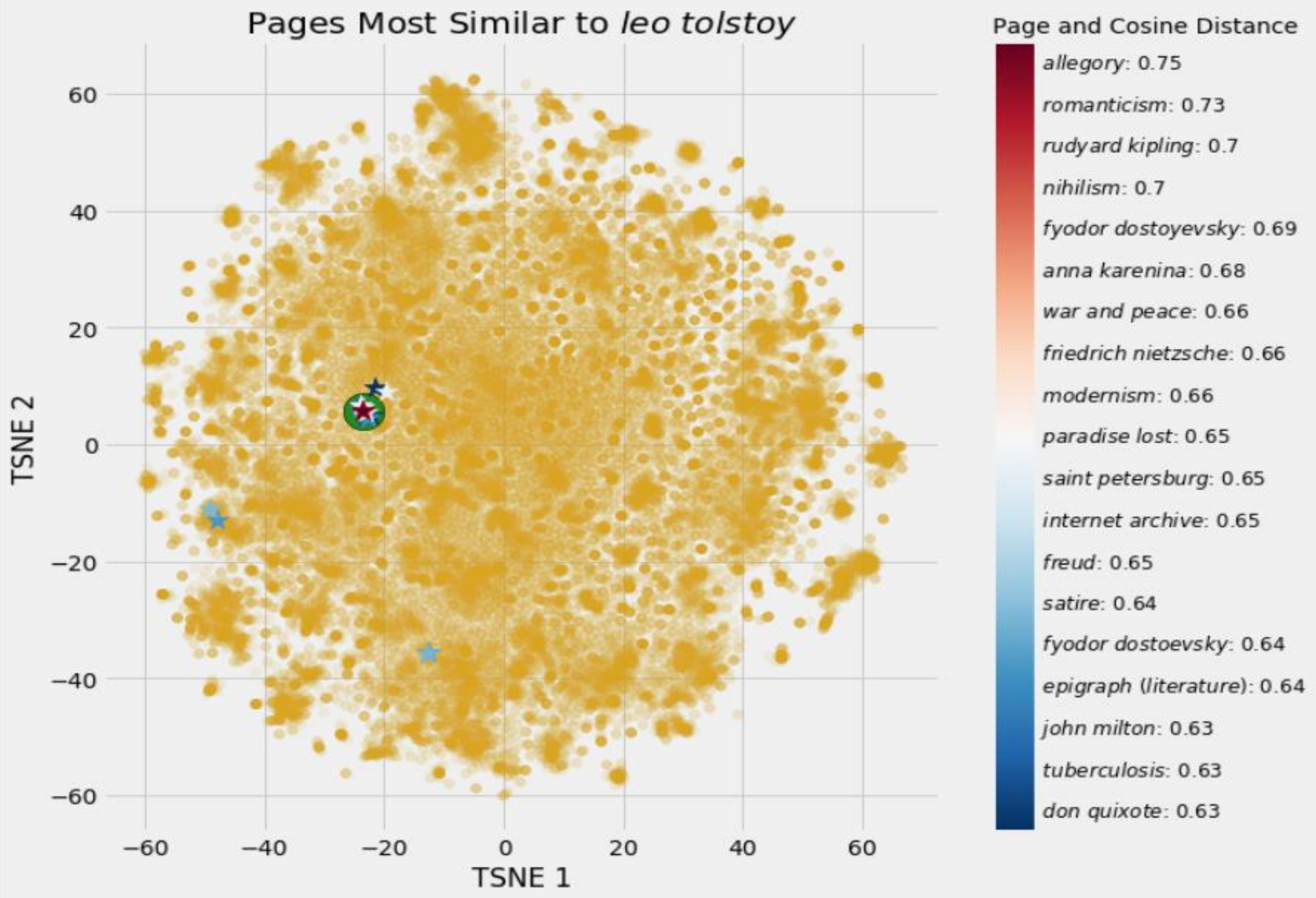
TSNE PAGES : JAME JOYCE 9 ITEMS



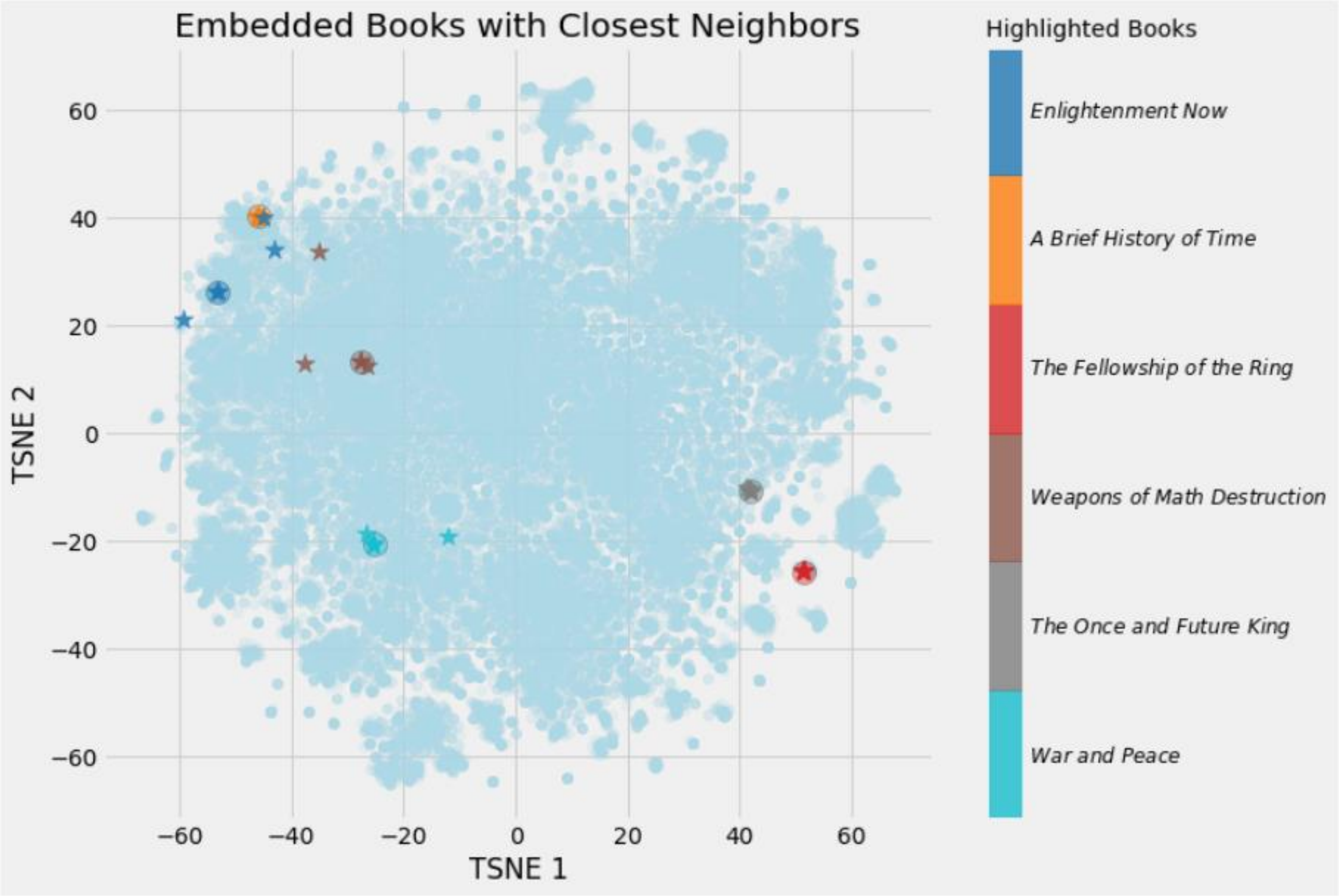
TSNE PAGES :MARGERET AT WOOD



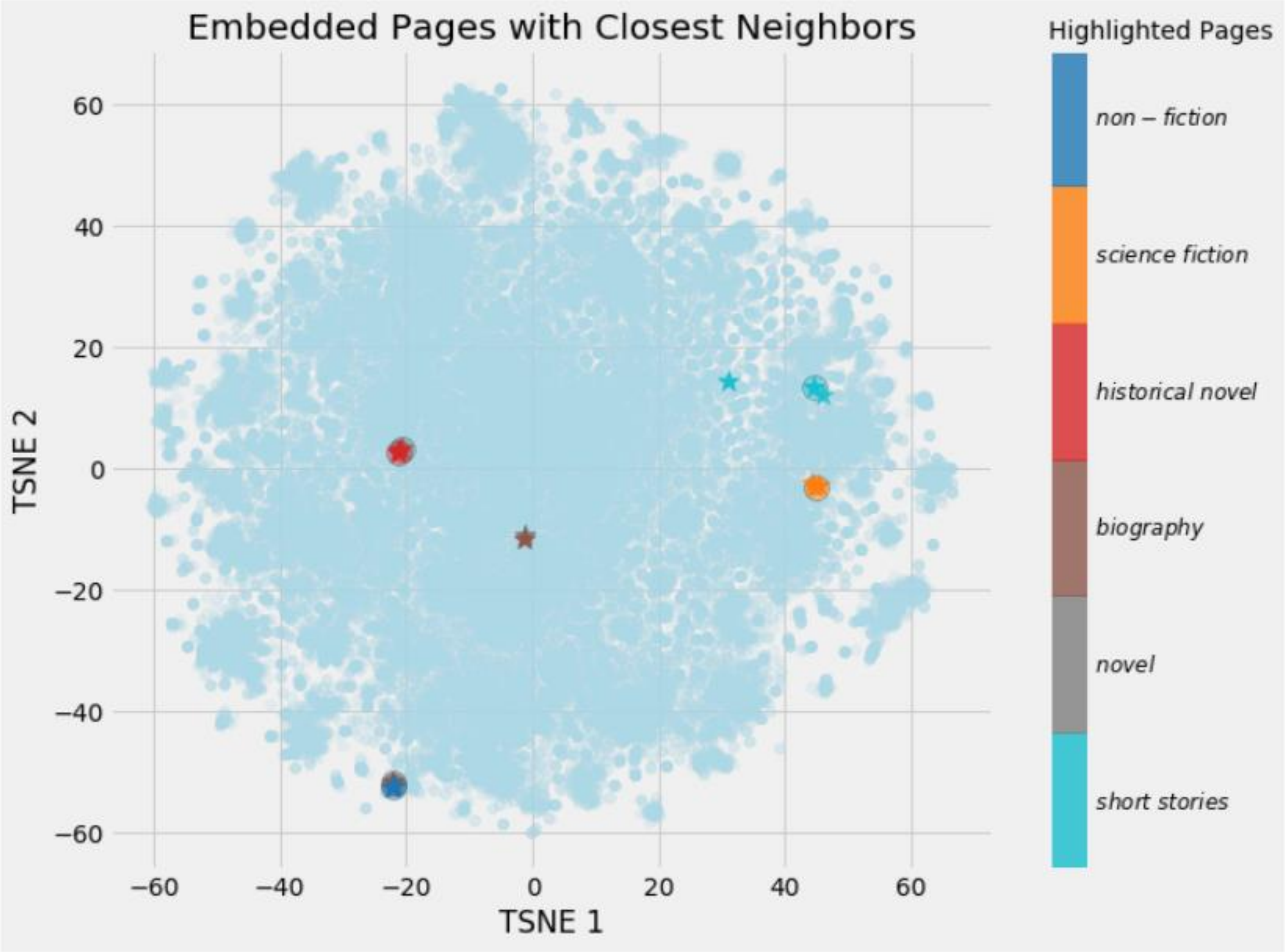
TSNE PAGES : LEO TOLSTOY



EMBEDED BOOK WITH CLOSEST NEIGHBORS



EMBEDED PAGES WITH CLOSEST NEIGHBORS




Thanks!
From Unsplash