

Document Similarity notebook – help pages

Introduction

The Document Similarity tool, as part of the [Australian Text Analytics Platform](#) (ATAP) project, is a Python package with a Jupyter notebook interface, developed by the [Sydney Informatics Hub](#) (SIH) in collaboration with the [Sydney Corpus Lab](#). It uses [MinHash](#) to efficiently estimate the [Jaccard similarity](#) between sets of English language documents.

(Note: if you are unfamiliar with how to use Jupyter Notebooks, have a look at this [guide](#).)

Getting started

The tool is available on [GitHub](#) where you can launch the tool as a Jupyter Notebook via Binder by clicking on one of the ‘launch binder’ buttons:

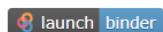
Setup

This tool has been designed for use with minimal setup from users. You are able to run it in the cloud and any dependencies with other packages will be installed for you automatically. In order to launch and use the tool, you just need to click the below icon.



Note: CILogon authentication is required. You can use your institutional, Google or Microsoft account to login.

If you do not have access to any of the above accounts, you can use the below link to access the tool (this is a free Binder version, limited to 2GB memory only).



It may take a few minutes for Binder to launch the notebook and install the dependencies for the tool. Please be patient.

The access to the ATAP Binderhub (i.e., the first ‘launch binder’ button) requires CILogon authentication, which supports single sign-on (SSO) method with most (Australian or international) institutional login credentials as well as some other accounts. If you have access to software that supports Jupyter Notebooks, you can also clone the Github repository and use the notebook locally (i.e., without Internet connection) on your own computer.

Overview of Tool

If you have already read [the blog post introducing this tool](#) or are familiar with the tool, you can skip this general overview section and will find the tool explanation from [here](#) onwards.

The tool is designed to compare every pair of texts in a dataset or corpus and produce similarity scores. Based on several customisable parameters (or the default values), the tool retrieves pairs of texts whose similarity scores exceed the pre-determined cut-off. The result

of this analysis is presented as a table containing a list of similar documents (in pairs) found by the tool – an example is shown in Table 1 below. Based on this analysis, the tool makes recommendations about which text within each similar pair should be kept or removed – as shown in the ‘status1’ and ‘status2’ columns in Table 1.

Table 1. Results of similarity analysis

	text_id1	text_name1	word_count1	status1	similarity	text_id2	text_name2	word_count2	status2
0	03eec57bba	text4	553	keep	1.0000	611849dd2b	text10	552	remove
1	2be0d88401	text5	202	remove	0.9922	124c11ee87	text2	200	remove
2	03eec57bba	text4	553	keep	0.9883	c756f03105	text7	555	keep
3	611849dd2b	text10	552	remove	0.9883	c756f03105	text7	555	keep
4	d726445d84	text6	92	keep	0.9805	752e10b6f7	text3	90	remove
5	35ea0dcfe0	text8	205	keep	0.9766	124c11ee87	text2	200	remove
6	d726445d84	text6	92	keep	0.9766	68bb00bb29	text9	92	remove
7	03eec57bba	text4	553	keep	0.9727	ee10095c6f	text1	547	remove

The tool allows you to view the content of selected similar documents (by specifying the row index you wish to analyse; see Figure 1 below), analyse them, and update the actions (i.e., ‘keep’ and ‘remove’). You can then download the non-duplicated texts (those labelled as ‘keep’) or the duplicated ones (those labelled as ‘remove’) into a zip archive of text (.txt) files.

Select row index:

Select action:

keep left text only

Display pair of texts

Update selection

Save table

Text: text4

Text: text10

text id: 03eec57bba; word count: 553; Jaccard similarity: 1.0; status: keep

text id: 611849dd2b; word count: 552; Jaccard similarity: 1.0; status: remove

Facebook and Instagram, which Facebook owns, followed up in the evening, announcing that Trump wouldn't be able to post for 24 hours following two violations of its policies

Facebook and Instagram, which Facebook owns, followed up in the evening, announcing that Trump wouldn't be able to post for 24 hours following two violations of its policies

The White House did not immediately offer a response to the actions

The White House did not immediately offer a response to the actions

While some cheered the platforms' response, experts noted that these actions follow years of hemming and hawing regarding Trump and his supporters spreading dangerous misinformation and encouraging

While some cheered the platforms' response, experts noted that these actions follow years of hemming and hawing regarding Trump and his supporters spreading dangerous misinformation and encouraging

Figure 1. Viewing and comparing each pair of texts in order to adjust the action (i.e., ‘keep’ or ‘remove’)

Additionally, the tool allows you to visualise the Jaccard similarity scores as a histogram (Figure 2). The histogram shows the Jaccard similarity scores for every pair of texts/documents in the corpus, and how many similar documents are found at those Jaccard similarity score ranges across the corpus. This is useful for estimating the extent of duplicated content in a corpus.

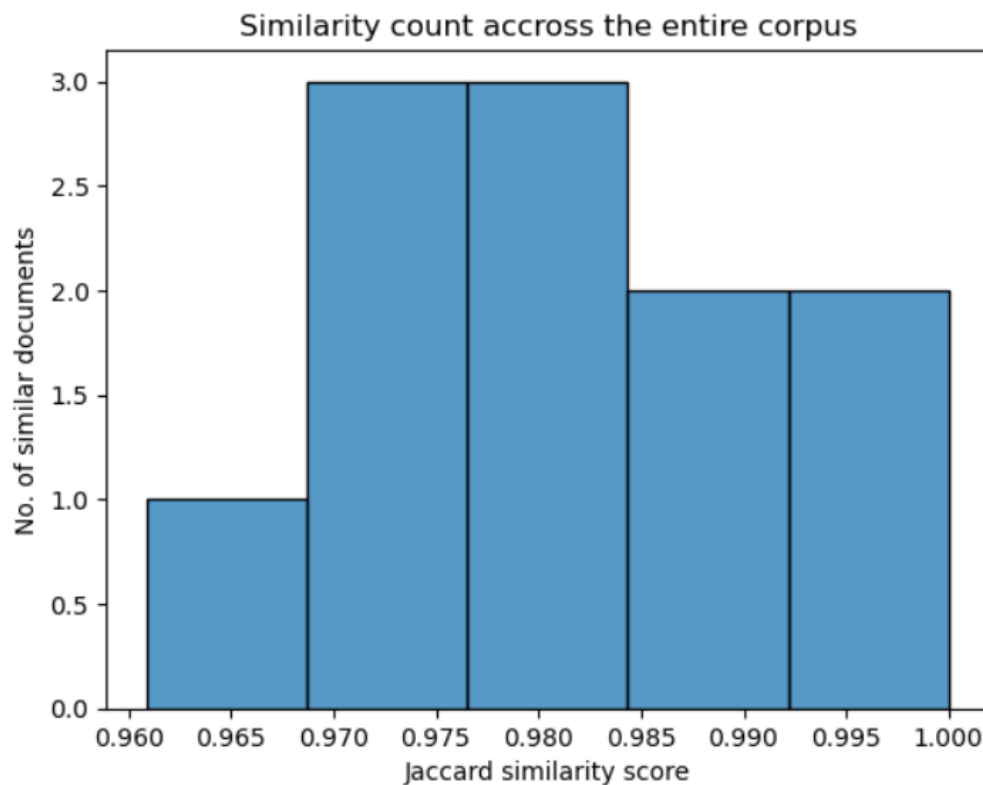


Figure 2. Histogram of the count of similar documents in the corpus found in particular Jaccard similarity score ranges.

Another, optional, visualisation shows the Jaccard similarity scores between specific pairs of texts/documents as a heatmap. This can be useful for identifying the texts that are most similar to each other but works best with small numbers of texts. This optional feature of the tool could be used to compare texts produced by different speakers/authors in a dataset.

Setup

Before you begin, you need to import the DocumentSimilarity package and the necessary libraries and initiate them to run in this notebook.

1. Execute the cell (as a reminder, you can do this by pressing 'Ctrl' + 'Enter' on PC or 'Cmd' + 'Enter' on Mac):


```
[ ]: # import the DocumentSimilarity tool
print('Loading DocumentSimilarity...')
from document_similarity import DocumentSimilarity, DownloadFileLink
import sys

# initialize the DocumentSimilarity
ds = DocumentSimilarity()
print('Finished loading.')
```

2. Once completed, you should get a message saying, "Finished loading":

```
[1]: # import the DocumentSimilarity tool
print('Loading DocumentSimilarity...')
from document_similarity import DocumentSimilarity, DownloadFileLink
import sys

# initialize the DocumentSimilarity
ds = DocumentSimilarity()
print('Finished loading.')

Loading DocumentSimilarity...
 BokehJS 2.4.3 successfully loaded.

Finished loading.
```

Load the data


This notebook will allow you to upload text data in a text file (or a number of text files) or a spreadsheet.

1. Execute the cell:

```
[ ]: # upload the text files and/or excel spreadsheets onto the system
display(ds.upload_box)
print('Uploading large files may take a while. Please be patient.')
print('\033[1mPlease wait and do not press any buttons until the progress bar appears...\033[0m')
```

2. Click 'Upload your files'. A window should appear prompting you to select txt files, or a single csv file, xlsx file, or zip folder.


```
[2]: # upload the text files and/or excel spreadsheets onto the system
display(ds.upload_box)
print('Uploading large files may take a while. Please be patient.')
print('\033[1mPlease wait and do not press any buttons until the progress bar appears...\033[0m')
```

 Upload your files (txt, csv, xlsx or zip) (0)

Uploading large files may take a while. Please be patient.
Please wait and do not press any buttons until the progress bar appears...

3. Click 'Open' after you've selected the file(s) you want to upload.
4. The tool should start loading the selected file(s). Once completed, you get a message saying, "Finished uploading files..." and another message saying the number of files that were uploaded (e.g., "10 text documents are loaded for tagging"):

```
[2]: # upload the text files and/or excel spreadsheets onto the system
display(ds.upload_box)
print('Uploading large files may take a while. Please be patient.')
print('\033[1mPlease wait and do not press any buttons until the progress bar appears...\033[0m')
```

 Upload your files (txt, csv, xlsx or zip) (0)

The total size of the upload is 0.02 MB.

Reading uploaded files...

This may take a while...

100%|██████████| 10/10 [00:00<00:00, 34017.06it/s]

Finished uploading files.

10 text documents are loaded for tagging.

rm: cannot remove './input': No such file or directory

Uploading large files may take a while. Please be patient.

Please wait and do not press any buttons until the progress bar appears...

5. Execute the following cell to view a snippet of the contents of the files you've uploaded:

```
[ ]: # display uploaded text
n=5

ds.text_df.head(n)
```

By default, you can view (up to) five of the files. You can adjust this by changing the number value in “n=5” code (to e.g., n=2, n=10, n=20) before you execute the cell.

```
[3]: # display uploaded text
n=5

ds.text_df.head(n)
```

```
[3]:
```

	text_name	text	text_id
0	text1	Facebook and Instagram, which Facebook owns, f...	ee10095c6f
1	text2	(CBC News)\nRepublican lawmakers and previous ...	124c11ee87
2	text3	Federated States of Micronesia President David...	752e10b6f7
3	text4	Facebook and Instagram, which Facebook owns, f...	03eec57bba
4	text5	(CBC News)\nRepublican lawmakers and previous ...	2be0d88401

Calculate Document Similarity

Once your texts have been uploaded, you can begin to calculate the similarity between documents in the corpus. The DocumentSimilarity tool uses Jaccard similarity to measure the similarity between documents.

1. You can change the variables/parameters (for calculating similarity) in the cell shown below:

```
[ ]: # USER SPECIFIES THESE VARIABLES
# set the n-gram size (the number of words used to detect similarity),
# e.g., n-gram=1 means compare every word ('apple' and 'orange'),
# n-gram=2 means compare every pair of words ('one apple' and 'two oranges'), etc.
ngram_value = 1

# select whether to calculate actual or estimated Jaccard similarity
# to measure the similarity between documents
# we recommend using estimated Jaccard similarity for large corpus of documents (faster)
actual_jaccard = False # True or False

# whether to exclude punctuations when calculating Jaccard similarity
ds.exclude_punc = False # True or False

# set the number of permutation functions (num_perm) parameter for estimating Jaccard similarity
# higher permutation functions improves the accuracy, but also increases query cost
num_perm = 256

# anything with >= the cutoff will be identified as similar documents
similarity_cutoff = 0.5 # value should be between 0-1
```

2. Set the *n*-gram (i.e., strings of adjacent word/s) size, which is used by the tool to detect similarity. You can do this by changing the number value in “ngram_value = 1”. For example, “ngram_value = 2” means compare every 2-word strings (e.g., “one apple” and “two oranges”), “ngram_value = 3” means every 3-word strings (e.g., “one green apple” and “two blood oranges”), and so on. The default setting is “ngram_value = 1” meaning that the tool compares every individual word (e.g., “apple” and “orange”).
3. Select whether to calculate actual or estimated Jaccard similarity to measure the similarity between documents. By default, the tool uses the “actual_jaccard = False” setting which means it will only estimate the Jaccard similarity with MinHash. To calculate actual Jaccard similarity scores on possible pairs, change the value from “False” to “True” (i.e., “actual_jaccard = True”).
4. Select whether to exclude punctuations when calculating Jaccard similarity. The tool will include punctuations by default (i.e., “ds.exclude_punc = False”). If you want the tool to ignore punctuations, change the value from “False” to “True” (i.e., “ds.exclude_punc = True”).
5. Set the number of permutation functions (num_perm) parameter for estimating Jaccard. Higher similarity permutation functions improve the accuracy, but increase the time needed to complete the calculation.
6. Set the similarity cut-off. This value needs to be between 0 and 1. The default cut-off is 0.5 (i.e., “similarity_cutoff = 0.5”). Any documents with similarity scores higher than or equal to the cut-off will be identified as similar documents.
7. Execute the cell to set the parameters for the similarity calculations.
8. Execute the following cell to begin the similarity calculations:

```
[ ]: # begin the process of calculating similarity and identify similar documents
ds.calculate_similarity(ngram_value, num_perm, similarity_cutoff, actual_jaccard)
```

9. Once the calculations are completed, if you get a message telling “No similar documents found. Please use lower similarity cutoff to find similar documents...”, it means no similar documents can be found in your corpus/dataset under the current settings/parameters. If there are similar documents, you will get a message telling you how many pairs of similar documents are in your corpus/dataset. For example:

```
[5]: # begin the process of calculating similarity and identify similar documents
ds.calculate_similarity(ngram_value, num_perm, similarity_cutoff, actual_jaccard)

12 pair of similar documents found in the corpus.
```

Please note that the following uses of the tool will only be possible if similar document pairs are identified in your corpus/dataset.

Analyse similar documents

Once the tool has finished calculating the document similarity, you can visualise and analyse the outcome.

Histogram of similar documents

You can use the tool to produce a histogram of the count of similar documents in the corpus as measured by their Jaccard similarity. Using the histogram, you can identify how many documents are found at different level of similarity measures.

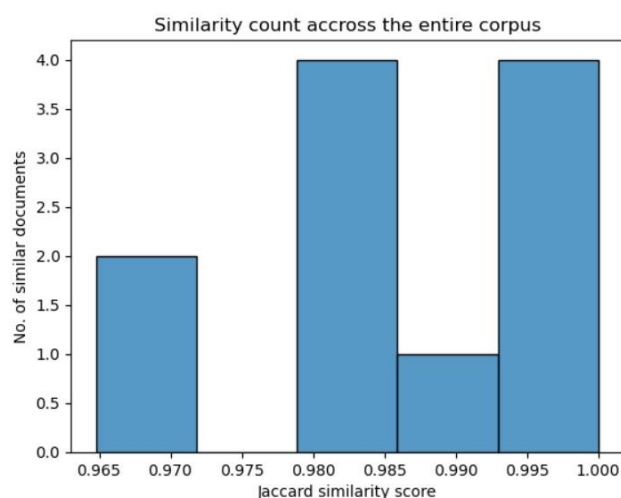
1. Execute the following cell:

```
[ ]: # plot the similarity count across the entire corpus
ds.plot_hash_similarity_by_source(ds.deduplication_df)
```

2. Once completed, you should get a histogram like the one below:

```
[6]: # plot the similarity count across the entire corpus
ds.plot_hash_similarity_by_source(ds.deduplication_df)

[6]: <AxesSubplot:title=('center': 'Similarity count across the entire corpus'), xlabel='Jaccard similarity score', ylabel='No. of similar documents'>
```



Heatmap of similar documents

You can produce a heatmap that shows the Jaccard similarity scores between pairs of similar documents. The heatmap only displays pairs of similar documents with scores above the similarity cut-off, as defined during the similarity calculations step.

1. You can set/change the settings (e.g., define the plot width, height, font size, and font colour) for the heatmap in the cell shown below:

```
[ ]: # define the plot width, height, font size and color
plot_width = 900 # increase plot width if necessary
plot_height = 800 # increase plot height if necessary
font_size = '14px'
text_color = 'white' # 'black' or 'white' would usually work for most scenarios

print('\033[1mVisualizing a large number of similar document pairs (>500) may slow down the notebook.\033[0m')
print('There are \033[1m{}\033[0m document pairs in the current process'.format(ds.deduplication_df.shape[0]))
plot_range = input("Enter the range of documents pairs to be plotted, e.g. y, n, 10-25, or 30.")

# plot heatmap of Jaccard similarity
ds.plot_heatmap_similarity(similarity_cutoff,
                           plot_width,
                           plot_height,
                           font_size,
                           text_color,
                           plot_range)
```

2. Set the plot width and plot height for the heatmap by changing the number value. By default, the size of the heatmap will be 900 x 800 pixels (i.e., “plot_width = 900” and “plot_height = 800”).
3. Set the font size for the size of the text for the title, axis labels (i.e., the name of the axis), the scale/units of the axis, and the Jaccard similarity score in the plot. The default font size is 14 (i.e., font_size = ‘14px’).
4. Set the font colour for the Jaccard similarity scores in the heatmap. White is the default font colour (text_color = ‘white’) with black being recommended as the alternate colour. However, you can use other CSS colour names (e.g., cyan, lime, pink, yellow; see [here](#) for the full list of colour names). Please note that if you enter an invalid colour (i.e., a name or word for a colour the tool does not recognise), then the Jaccard similarity scores will not appear in the resulting heatmap.
5. Execute the cell once you’re happy with the settings.
6. Specify the range of matched documents to be included in the heatmap:


```
[*]: # define the plot width, height, font size and color
plot_width = 900 # increase plot width if necessary
plot_height = 800 # increase plot height if necessary
font_size = '14px'
text_color = 'white' # 'black' or 'white' would usually work for most scenarios

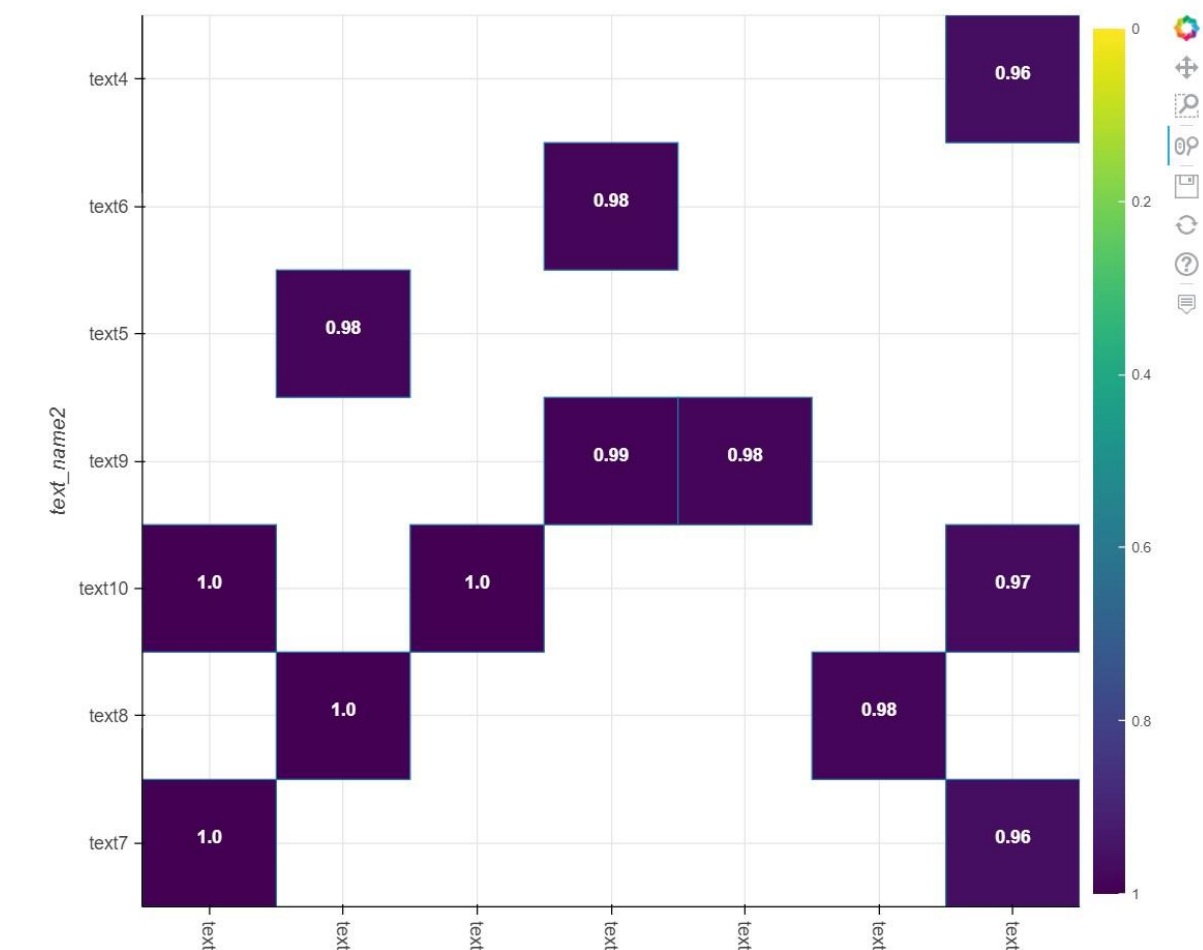
print('\033[1mVisualizing a large number of similar document pairs (>500) may slow down the notebook.\033[0m')
print('There are \033[1m{}\033[0m document pairs in the current process'.format(ds.deduplication_df.shape[0]))
plot_range = input("Enter the range of documents pairs to be plotted, e.g. y, n, 10-25, or 30.")

# plot heatmap of Jaccard similarity
ds.plot_heatmap_similarity(similarity_cutoff,
                          plot_width,
                          plot_height,
                          font_size,
                          text_color,
                          plot_range)
```

Visualizing a large number of similar document pairs (>500) may slow down the notebook.
 There are 12 document pairs in the current process
 Enter the range of documents pairs to be plotted, e.g. y, n, 10-25, or 30.

Entering an integer number, such as “30”, will generate the figure with the top-30 pairs of the similar documents. Entering a number range like “15-45” will generate the figure with the selected range (i.e., the 15th to the 45th) of the document pairs. Entering “y” will proceed with all pairs of similar documents. Entering “n” will cancel the figure generation.

- Click ‘Enter’ (on your keyboard) once you’ve entered the value you want, and the tool will produce a heatmap:



The x-axis and y-axis show the text_id of the similar document pairs (you can hover over the similar nodes to display the text name pairs in the tooltip). You can also zoom in/out of the

heatmap, move it around, download/save the visualisation to your local computer using the interactive toolbar on the right hand-side of the heatmap.

Analyse similar documents

You can generate a list of similar documents (in pairs) found by the tool, based on the similarity cutoff specified earlier. By default, the tool makes recommendations on whether to “keep” or “remove” each similar document within each similar pair. You can view each pair of similar documents (by specifying the row index you wish to analyse), and then update the action/recommendation (i.e., “keep” or “remove”) accordingly.

1. Execute the cell shown below:

```
[ ]: ds.display_deduplication_text()
```

2. The tool should produce a table that displays texts identified as similar based on the Jaccard similarity cut-off selected earlier. For example:

```
[8]: ds.display_deduplication_text()
```

	text_id1	text_name1	word_count1	status1	similarity	text_id2	text_name2	word_count2	status2
0	03eec57bba	text4	553	remove	1.0000	c756f03105	text7	555	keep
1	124c11ee87	text2	200	remove	0.9961	35ea0dcfe0	text8	205	keep
2	03eec57bba	text4	553	remove	0.9961	611849dd2b	text10	552	remove
3	c756f03105	text7	555	keep	0.9961	611849dd2b	text10	552	remove
4	752e10b6f7	text3	90	remove	0.9883	68bb00bb29	text9	92	remove
5	124c11ee87	text2	200	remove	0.9844	2be0d88401	text5	202	remove
6	d726445d84	text6	92	keep	0.9844	68bb00bb29	text9	92	remove

Select row index:

Select action:



Display pair of texts

Update selection

Previous pair

Next pair

Save table

3. Click on “Display pair of texts” to view each pair of similar texts.

[8]: ds.display_deduplication_text()

[8]:

	text_id1	text_name1	word_count1	status1	similarity	text_id2	text_name2	word_count2	status2
0	03eec57bba	text4	553	remove	1.0000	c756f03105	text7	555	keep
1	124c11ee87	text2	200	remove	0.9961	35ea0dcfe0	text8	205	keep
2	03eec57bba	text4	553	remove	0.9961	611849dd2b	text10	552	remove
3	c756f03105	text7	555	keep	0.9961	611849dd2b	text10	552	remove
4	752e10b6f7	text3	90	remove	0.9883	68bb00bb29	text9	92	remove
5	124c11ee87	text2	200	remove	0.9844	2be0d88401	text5	202	remove
6	d726445d84	text6	92	keep	0.9844	68bb00bb29	text9	92	remove

Select row index: Select action: keep right text only

Display pair of texts Update selection

Previous pair Next pair Save table

Text: text4

text id: 03eec57bba; word count: 553; Jaccard similarity: 1.0; status: remove

Facebook and Instagram, which Facebook owns, followed up in the evening, announcing that Trump wouldn't be able to post for 24 hours following two violations of its policies

Text: text7

text id: c756f03105; word count: 555; Jaccard similarity: 1.0; status: keep

Facebook and Instagram, which Facebook owns, followed up in the evening, announcing that Trump wouldn't be able to post for 24 hours following two violations of its policies

To view different pairs of similar texts, click “Previous pair” or “Next pair”. To view a specific pair, type the corresponding row number (for the pair of texts you wish to view) into the box next to “Select row index”.

Alternatively, you can type a row number into “Select row index” to view a specific row. By default, the tool will show the first pair which is in row 0. Replace the 0 with 1 to view the second pair of texts, replace with 2 to view the third pair, and so on.

When you are viewing the pair of texts side by side, you may see text highlighted in yellow. These are chunks of texts that are not identical within the identical pair. For example:

Text: text3

text id: 752e10b6f7; word count: 90; Jaccard similarity: 0.9883; status: remove

Federated States of Micronesia President David Panuelo warned the deal could stoke geopolitical tensions and undermine the sovereignty of the Pacific, while Samoa's Prime Minister, Fiame Naomi Mata'afa, suggested the process had been rushed

Pacific expert Anna Powles from Massey University told the ABC “very little” was currently known about the meeting but it seemed to be part of China's broader attempts to sideline the PIF

“It appears to be an attempt to deliberately disrupt existing regional mechanisms which China is not a part of,” she said

Text: text9

text id: 68bb00bb29; word count: 92; Jaccard similarity: 0.9883; status: remove

Federated States of Micronesia President David Panuelo warned the deal could stoke geopolitical tensions and undermine the sovereignty of the Pacific, while Samoa's Prime Minister, Fiame Naomi Mata'afa, suggested the process had been rushed

Pacific expert Anna Powles from Massey University told the ABC “very little” was currently known about the meeting but it seemed to be part of China's broader attempts to sideline the PIF

“It appears to be an attempt to deliberately disrupt existing regional mechanisms which China is not a part of,” she said this afternoon

- After viewing the similar pair of texts, click on the drop-down menu next to “Select action” to update the status (i.e., either to “keep” or “remove”) for each text in the pair:

```
[8]: ds.display_deduplication_text()
```

[8]:

	text_id1	text_name1	word_count1	status1	similarity	text_id2	text_name2	word_count2	status2
0	03eec57bba	text4	553	remove	1.0000	c756f03105	text7	555	keep
1	124c11ee87	text2	200	remove	0.9961	35ea0dcfe0	text8	205	keep
2	03eec57bba	text4	553	remove	0.9961	611849dd2b	text10	552	remove
3	c756f03105	text7	555	keep	0.9961	611849dd2b	text10	552	remove
4	752e10b6f7	text3	90	remove	0.9883	68bb00bb29	text9	92	remove
5	124c11ee87	text2	200	remove	0.9844	2be0d88401	text5	202	remove
6	d726445d84	text6	92	keep	0.9844	68bb00bb29	text9	92	remove

Select row index:

Select action: keep right text only ▼

Display pair of texts Update selection

Previous pair Next pair Save table

As you can see in the image above, you have the option to “keep left text only” (i.e., status1 is “keep”, status2 is “remove”), “keep right text only” (i.e., status1 is “remove”, status2 is “keep”), “keep both” (i.e., status1 and status2 are “keep”), or “remove both” (i.e., status1 and status2 are “remove”).

5. You can save the table by clicking on “Save table”. Click on the link to a csv file that appears to download and save it to your computer:

[8]:

	text_id1	text_name1	word_count1	status1	similarity	text_id2	text_name2	word_count2	status2
0	03eec57bba	text4	553	remove	1.0000	c756f03105	text7	555	keep
1	124c11ee87	text2	200	remove	0.9961	35ea0dcfe0	text8	205	keep
2	03eec57bba	text4	553	remove	0.9961	611849dd2b	text10	552	remove
3	c756f03105	text7	555	keep	0.9961	611849dd2b	text10	552	remove
4	752e10b6f7	text3	90	remove	0.9883	68bb00bb29	text9	92	remove
5	124c11ee87	text2	200	remove	0.9844	2be0d88401	text5	202	remove
6	d726445d84	text6	92	keep	0.9844	68bb00bb29	text9	92	remove

Select row index:

Select action: keep right text only ▼

Display pair of texts Update selection

Previous pair Next pair Save table

Table saved. Click below to download:
[deduplication_table.csv](#)

Save duplicated/non-duplicated texts

Once you are happy with the list of texts that you want to keep, you can save the non-duplicated texts (those with “keep” status) or the duplicated ones (those with “remove” status) as a zip of text (.txt) files and download them to your local computer.

1. Execute the cell shown below to preview the texts you are keeping:

```
[ ]: rows_to_display=5  
ds.finalise_and_save(rows_to_display)
```

You can adjust how many texts you can preview by changing the number value next to “rows_to_display”. By default, you can view five texts (i.e., “rows_to_display=5”).

2. A table containing the texts you are keeping (i.e., the ones with a “keep” status) should appear:

```
[9]: rows_to_display=5  
ds.finalise_and_save(rows_to_display)
```

	text_name	text	text_id	word_count
3	text4	Facebook and Instagram, which Facebook owns, f...	03eec57bba	553
4	text5	(CBC News)\nRepublican lawmakers and previous ...	2be0d88401	202
5	text6	Federated States of Micronesia President David...	d726445d84	92
6	text7	Facebook and Instagram, which Facebook owns, f...	c756f03105	555
7	text8	(CBC News)\nRepublican lawmakers and previous ...	35ea0dcfe0	205

Save non-duplicated texts

Save duplicated texts

3. Click on “Save non-duplicated texts” to download and save the texts you are keeping for the final version of your corpus/dataset. Click on the link to a zip folder containing all of these texts that appears to start the download to your computer:

[9]:

	text_name	text	text_id	word_count
3	text4	Facebook and Instagram, which Facebook owns, f...	03eec57bba	553
4	text5	(CBC News)\nRepublican lawmakers and previous ...	2be0d88401	202
5	text6	Federated States of Micronesia President David...	d726445d84	92
6	text7	Facebook and Instagram, which Facebook owns, f...	c756f03105	555
7	text8	(CBC News)\nRepublican lawmakers and previous ...	35ea0dcfe0	205

Save non-duplicated texts

100% | 5/5 [00:00<00:00, 1442.53it/s]

Your texts have been saved. Click below to download:

[deduplicated_texts.zip](#)

Save duplicated texts

- Click on “Save duplicated texts” to download and save the texts you are removing from your corpus/dataset (i.e., the ones with a “remove” status). Click on the link to download a zip archive that contains all of the selected texts to your computer:

Save duplicated texts

100% | 5/5 [00:00<00:00, 1104.00it/s]

Your texts have been saved. Click below to download:

[duplicated_texts.zip](#)

Once you’re done with the notebook, make sure you shut down the kernel. As a reminder, to shut down a kernel, go to click on ‘Kernel’ (in the menu) then ‘Shut Down Kernel’ (or ‘Shut Down All Kernels...’).

Citing/Referencing this Notebook

Citation: Jufri, Sony & Sun, Chao (2022). Document Similarity. v1.0. Australian Text Analytics Platform. Software. <https://github.com/Australian-Text-Analytics-Platform/document-similarity>

If you are using this notebook in your research, please also include the following statement or an appropriate variation thereof:

This study has utilised a notebook/notebooks developed for the Australian Text Analytics Platform (<https://www.atap.edu.au>) available at <https://github.com/Australian-Text-Analytics-Platform/document-similarity/>.

In addition, please inform ATAP (info@atap.edu.au) of publications and grant applications deriving from the use of any ATAP notebooks in order to support continued funding and development of the platform.

Acknowledgments

This Jupyter notebook and relevant python scripts were developed by the Sydney Informatics Hub (SIH) in collaboration with the Sydney Corpus Lab under the [Australian Text Analytics Platform program](#) and the [HASS Research Data Commons and Indigenous Research Capability Program](#). These projects received investment from the Australian Research Data Commons ([ARDC](#)), which is funded by the National Collaborative Research Infrastructure Strategy ([NCRIS](#)).

The notebook incorporates MinHash, which is introduced by Andrei Z. Broder in this [paper](#). Details can be found [here](#).

Known Issues

The notebook has not been tested with very big data sets.