**Backed communication with user interface**

The input from the user would be saved to a configuration file as a *config.csv* as a comma-separated file. The config.csv would be input into my application as backend. In the following the contains of config.csv is described. Some sample data is defined for reference.

***config.csv***

The file is a comma-separated file with the following content:

*folder\_path ,* 'c:/trainingresource'

*proj\_dir, 'Docs with Acquisition Recommendations'*

*file\_bench, 'phil-bench.csv'*

*stop\_words\_file, 'stopwords\_en.txt'*

*max\_num\_files, 2000*

*threshold, .5*

*mav\_roll, 20*

The definition of each row would be as follows:

folder\_path , 'c:/trainingresource'

Input file directory. Define the directory path to the folder of the training dataset containing the entire pdf format file, which is used for training the model. All the subdirectories are read for finding the pdf files.

*proj\_dir, 'Docs with Acquisition Recommendations'*

Input file directory. The folder contains the pdf file which the user wants to examine their similarity to the benchmark set. Each pdf file in this folder has been read and compared to the benchmark and would be used for reporting its similarity. As result, their file name will be used in the reports as their representation.

*file\_bench, 'phil-bench.csv'*

Input file. This is a benchmark file containing the suggested benchmarks by the subject matter experts (SME). This file could be essentially create by our team and have a fixed content. The file is comma-separated, and it is shown partially in the following:

|  |  |  |
| --- | --- | --- |
| Benchmarks | weight | type |
| accurate | 1 |  |
| achievable | 1 |  |
| achieve | 1 |  |
| act | 2 |  |
| address | 2 |  |
| adhere | 3 | action |
| advantages | 1 |  |
| advise | 3 | action |
| affordable | 4 | objective |
| align | 3 | action |
| align | 3 | action |
| analytical | 1 |  |
| analyze | 3 | action |
| assess | 3 | action |
| assessment | 2 |  |
| assignment | 1 |  |
| avoid | 4 | stop or avoid |

The first column is the benchmark word or phrase. The second column is a measure between 1-5 which presents the importance of the benchmark. This value is used as a weight for finding the similarity of the documents’ words to that specific benchmark. The third column is the type of benchmark. It is not used yet and not filled for all benchmarks. However, it is suggested by the SMEs. Maybe it will be useable for categorization later. I have created the benchmark table with the help of SMEs and exploration of some recommendation reports to find the essential words and phrases indicating the recommendation intuition. The words without the type are my suggestions. The weight columns are adjusted from the subject matter expert on a scale of 1-5. They used another scale.

stop\_words\_file,'stopwords\_en.txt'

Input file. This file contains the stop words and is used by model.

*max\_num\_files, 2000*

Hyperparameter. This hyperparameter is the maximum number of files that would be read for training in the training phase. This parameter helps to limit number of input files. The input directory could contain a large number of files and the training time and memory use completely depend on this measure. About 1600 files with 3.5-gigabyte volume are currently used for training.

*threshold, .5*

Hyperparameter. This hyperparameter is used for comparing the similarity of the benchmarks and each word in a specific document. Similarity more than *threshold* would be counted toward similarity only. Increasing this threshold change the result significantly. It seems that 0.5 is an ideal value for this parameter.

*mav\_roll,20*

Hyperparameter. This hyperparameter is used for finding the specific part of the document which contained a recommendation. For this purpose, a moving average with *mav\_roll* rolling value is used to examine the entire document and find the parts that have a better chance of being a recommendation phrase. Currently, 20 is used and has an acceptable result.

**Produced outputs:**

'word2vec\_added.model'

Output file. This file is created during the training phase and will be used later in evaluating the input files. This default file name could be used if it is not included in config file. It is a *genism* model file that is created during training phase and load later for comparison by following statements:

*from gensim.models import Word2Vec*

*w2vec\_model = Word2Vec(min\_count=10,*

*window=7,*

*vector\_size=300,*

*sample=6e-5,*

*alpha=0.03,*

*min\_alpha=0.0007,*

*negative=20,*

*workers=cores-1)*

*w2vec\_model.build\_vocab(docs, progress\_per=10000)*

*w2vec\_model.train(docs, total\_examples=w2vec\_model.corpus\_count,*

*epochs=50, report\_delay=1.0*

*)w2vec\_model.save(model\_file)*

*.*

*.*

*.*

*w2vec=Word2Vec.load(model\_file)*

The *doc* is contained entire training corpus dataset in above codes.

'sim\_to\_each\_other.csv'

Output file. This file is created during examination of the input file. It contains a similarity measure between 0-1 which shows how similar the input pdf files are to each other. Each row is one file, and each column is also one file from the *proj\_dir.* The best format for showing the result is a heatmap as it is presented in the following figure *.*



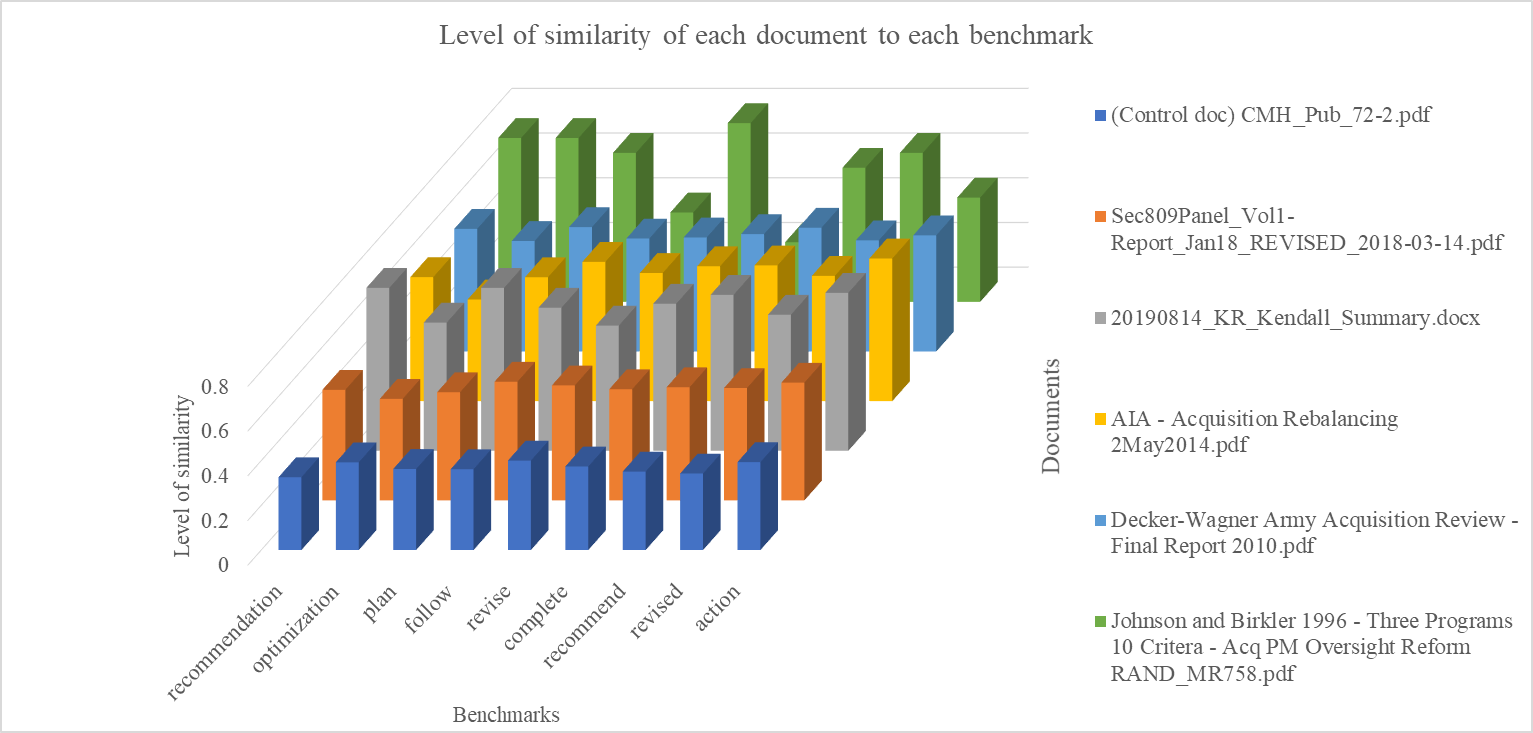
*'sim\_to\_benchmarks.csv'*

Output file. This file contains a matrix in which each row is a file from *proj\_dir* that shows how similar a specific pdf file is to the entire benchmark generally. The best format for showing this output is a heatmap again.



**'sim\_to\_benchmarks\_seperated.csv'**

Output file. This file contains a matrix in which each column is a file from *proj\_dir,* and each row is a specific benchmark. It shows how similar a particular pdf file is to each benchmark. The best format for showing this is a 3d bar chart with a sample selection of the benchmarks and sample selection of the pdf files.



**“SIMS”**

Output directory. This directory has been created, and for each project pdf file in *proj\_dir* a CSV file with the same file name would be created. Each file contains a table which format is presented as the following partially.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| doc\_words | similarity\_to\_bench | weight | W\_simil | MAV20 |
| getting | 0.22 | 2 | 0.44 | 0.48 |
| defense | 0.44 | 3 | 1.31 | 0.49 |
| acquisition | 0.60 | 1 | 0.60 | 0.48 |
| right | 0.31 | 3 | 0.92 | 0.48 |
| summary | 0.35 | 3 | 1.04 | 0.48 |
| improvement | 0.53 | 4 | 2.12 | 0.47 |
| set | 1.00 | 3 | 3.00 | 0.43 |
| reasonable | 0.37 | 1 | 0.37 | 0.43 |
| professional | 0.26 | 3 | 0.78 | 0.46 |
| resource | 1.00 | 3 | 3.00 | 0.41 |
| provide | 0.54 | 3 | 1.63 | 0.40 |
| strong | 0.27 | 2 | 0.54 | 0.41 |
| incentive | 0.52 | 4 | 2.07 | 0.40 |
| better | 1.00 | 2 | 2.00 | 0.37 |
| buying | 0.28 | 2 | 0.56 | 0.37 |
| power | 0.21 | 1 | 0.21 | 0.44 |
| principle | 0.36 | 1 | 0.36 | 0.42 |
| suggested | 0.19 | 3 | 0.57 | 0.48 |
| acquisition | 0.60 | 1 | 0.60 | 0.47 |
| principle | 0.36 | 1 | 0.36 | 0.46 |
| continuous | 0.27 | 4 | 1.06 | 0.47 |
| improvement | 0.53 | 4 | 2.12 | 0.45 |
| effective | 0.41 | 2 | 0.81 | 0.45 |
| radical | 0.21 | 3 | 0.62 | 0.47 |
| principle | 0.36 | 1 | 0.36 | 0.47 |

Each row in the table is the words in the document with a natural sequence that appears in the main document. The columns describe as follows.

*Doc\_words*

This column presents each word in the document.

*similarity\_to\_bench*

This column gives a measure between 0-1 that shows the similarity of the word to the entire benchmark. The value is calculated by finding the most similar benchmark to this word (max-pooling).

*Weight*

This column presents the weight of the benchmark, which is most similar to the word.

*W\_simil*

This parameter calculates the weighted similarity by using *Weight* And *similarity\_to\_bench.*

*MAV20*

This column presents a weighted moving average of the similarity for the word sequence. The rolling for this moving average would be the hyperparameter *mav\_roll.* The rolling is moving forward. The best way to show the result is a heatmap that covers the words in the document, as shown in following.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0.48 | 0.49 | 0.48 | 0.48 | 0.48 | 0.47 | 0.43 | 0.43 | 0.46 | 0.41 | 0.40 | 0.41 |
| getting | defense | acquisition | right | summary | improvement | set | reasonable | professional | resource | provide | strong |

The coloring code shows the content of the document in a format that explains how the phrases (word sequences with a length of rolling, which by default is 20) could be a recommendation phrase. Each MAV20 value shows how the following 20 words are like the benchmarks. This measure can represent the part of the documents that have more potential to be a recommendation phrase. The entire document could be color-coded with this algorithm, and the user can visually look at it and swiftly go through the most similar part to benchmarks.

**Directory structure:**

Following directory structure is used for the backend programs and datasets. The application python file would be at the workdirectory.

workdirectory-|

              |-trainingresource-|

              |                     |- pdf-dir1

              |                       |- pdf-dir2

           |- docs with Acquisition Recommendations

              |- SIMS

              |- \*.py