CREA_library Methods

Method	Description	Parameters	Output
`CREA_library()`	Create an instance of CREA_library	`dict`	None
`get_all_vectors()`	Get all vectors from dataset.	None	`dict`: All word vectors
`get_vector(word)`	Gets the vector for a specific word.	`word (str)`: Target word	`list` or `None`: Word vector if found
`get_vectors(words)`	Gets the vectors for a list of words.	`words (list)`: List of words	`dict`: Word-vector pairs for found words
`select_cols(words, columns)`	Selects specific columns from word vectors.	`words (list)`, `columns (list)`	`dict`: Words with selected columns
`cosine_similarity(vec1, vec2)`	Computes cosine similarity between two words or vectors.	`vec1 (str or list)`, `vec2 (str or list)`	`float`: Cosine similarity score
`top_n_similar(word, n=5)`	Finds the top `n` most similar words to a given word. Default 5	`word (str)`, `n (int, default=5)`	`list`: Top `n` most similar words with scores
`strip(file_name)`	Removes category names from a JSON file, returning float-only vectors.	`file_name (str)`: JSON file path	`list`: Processed word vectors
`get_raw(file_name)`	Extracts raw CREA dataset into a CSV file.	`file_name (str)`: JSON file path	Writes to `results.csv`
`load_word_from_json(file_name)`	Loads a dictionary of word vectors from a JSON file.	`file_name (str)`: JSON file path	`dict`: Word vectors
`calculate_averages(file_name)`	Calculate the averages of the CREA dataset	`file_name (str)`: CSV file path	`df`: Averages calculated from raw scores

strip is useful if the vectors have the category tags with them. For example:

```
{'molecule': {'Vision': 0.53125,
  'Bright': 0.96875,
  'Dark': 0.71875,
  'Color': 1.1875,
  'Pattern': 1.625,
  'Large': 0.125},

 'automobile: {'Vision': 5.53,
  'Bright': 1.9,
  'Dark': 1.83,
  'Color': 2.13,
  'Pattern': 2.7,
  'Large': 4.43 }}
```

Can be changed to:

```
{'molecule': [0.53125, 0.96875, 0.71875, 1.1875, 1.625, 0.125],
 'automobile': [5.53, 1.9, 1.83, 2.13, 2.7, 4.43]}
```

This can then be initialized by CREA_library. Now the remaining methods can be used. Examples:

```
stripped_dict = CREA_library.strip('file_path.json')

word_dict = CREA_library(stripped_dict)

word_dict.get_all_vectors()

single_word = word_dict.get_vector('word1')

multiple_words = word_dict.get_vectors(['word1', 'word2', ... , 'wordn'])

specific_cols = word_dict.select_cols(['word1'], [0, 1, 2])

similar = word_dict.cosine_similarity('word1', 'word2')

top_similar = word_dict.top_n_similar('word1', 5)
```

If getting data from psychopy and getting raw df or calculate averages

```
fname = 'results.json'
CREA_library.get_raw(fname)

CREA_library.calculate_averages('results.csv')
```

Exports data from psychopy named ‘results.json’ to a csv file that it updated with each new output of the experiment