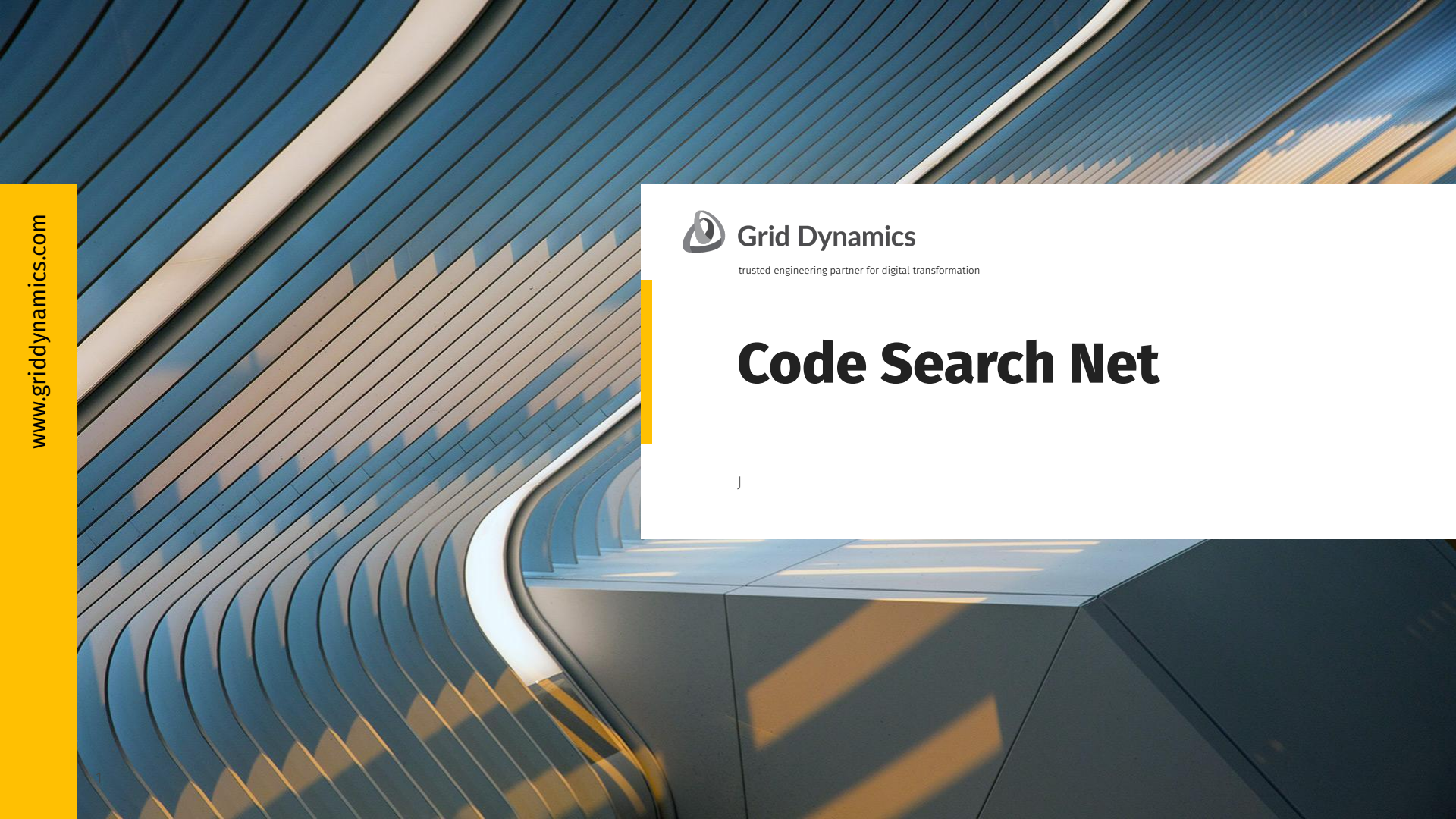


www.griddynamics.com

Grid Dynamics

trusted engineering partner for digital transformation

# Code Search Net

J

# Exploring the dataset

In our Dataset 1,50,000 rows and 12 columns are present . Columns are dosctring , code ,github link, url ,language, function name . original_string and code column are same , language in the dataset is python only , code_tokens and docstring_tokens are not required as processing to be done on docstring. Only useful columns for our use case is only docstring and code , with zero null value present in the dataset.

## Example

| func_name | original_string | language | code | code_tokens | docstring | docstring_tokens |
|---|---|---|---|---|---|---|
| pipe | def pipe(*args):\n """\n Takes as parame... | python | def pipe(*args):\n """\n Takes as parame... | [def, pipe, (, *, args, ), :, if, len, (, args... | Takes as parameters several dicts, each with t... | [Takes, as, parameters, several, dicts, each, ... |
| GdsLibrary.top_level | def top_level(self):\n """\n Out... | python | def top_level(self):\n """\n Out... | [def, top_level, (, self, ), :, top, =, list, ... | Output the top level cells from the GDSII data... | [Output, the, top, level, cells, from, the, GD... |
| synchronize | def synchronize():\n """\n Helper functi... | python | def synchronize():\n """\n Helper functi... | [def, synchronize, (, ), :, if, not, dist, ., ... | Helper function to synchronize (barrier) among... | [Helper, function, to, synchronize, (, barrier... |
| _get_data_versions | def _get_data_versions(data):\n """Retrieve... | python | def _get_data_versions(data):\n """Retrieve... | [def, _get_data_versions, (, data, ), :, genom... | Retrieve CSV file with version information for... | [Retrieve, CSV, file, with, version, informati... |
| HistoryAwareReferenceField.retrieve_version | def retrieve_version(self, obj, version):\n ... | python | def retrieve_version(self, obj, version):\n ... | [def, retrieve_version, (, self, ,, obj, ,, ve... | Retrieve the version of the object | [Retrieve, the, version, of, the, object] |

# Removing all rows with a language other than English.

- Docstring is present  in more than 1 language.
- We want only English as a language.
- Removing all the rows that has language other than English.
- After removing all the other languages the size of dataset is 1,15,643.

| | | |
|---|---|---|
| **7991** | Returns an optional configuration value, as an... | def get_int(self, key: str) -> Optional[int]:\... |
| **64640** | Touch every point on an object 'numInitialTrav... | def doExperiment(numColumns, l2Overrides, obje... |
| **53573** | 为task新建一个后台下载线程, 并开始下载. | def start_worker(self, row):\n '''为task... |
| **121907** | Test whether a key is a label reference for a ... | def _is_label_reference(self, key, axis=0):\n ... |
| **7031** | move source to destination. Can handle uploadi... | def move_to_destination(source, destination, j... |

# Processing the Data

- Converting all text in the lower case.
- Removing all numeric and extra spaces in the text.
- Removing all stop word like (is , the , a ) from english vocabulary.
- Lemmatizing the word like converting sums -> sum with proper meaning in English vocabulary
- Tokenizing the text and making new column with these tokenized docstring.
- After doing tokenization  removing all the rows that has size of 2 or less than 2. The shape of dataset after doing the processing is  [113884 ,  4] .

| | docstring | code | tokenized_docstring | tokenized_code |
|---|---|---|---|---|
| 0 | Cleanly shutdown the router socket | def close(self):\n '''\n Cleanly... | cleanly shutdown router socket | def close self cleanly shutdown router socket ... |
| 1 | Pre-fork we need to create the zmq router devi... | def pre_fork(self, process_manager):\n ... | pre fork need create zmq router device param f... | def pre_fork self process_manager pre fork nee... |
| 2 | After forking we need to create all of the loc... | def post_fork(self, payload_handler, io_loop):... | forking need create local socket listen router... | def post_fork self payload_handler io_loop for... |
| 3 | Handle incoming messages from underlying TCP s... | def handle_message(self, stream, payload):\n ... | handle incoming message underlying tcp stream ... | def handle_message self stream payload handle ... |
| 4 | Bind to the interface specified in the configu... | def _publish_daemon(self, log_queue=None):\n ... | bind interface specified configuration file | def _publish_daemon self log_queue none bind i... |

# Embedding

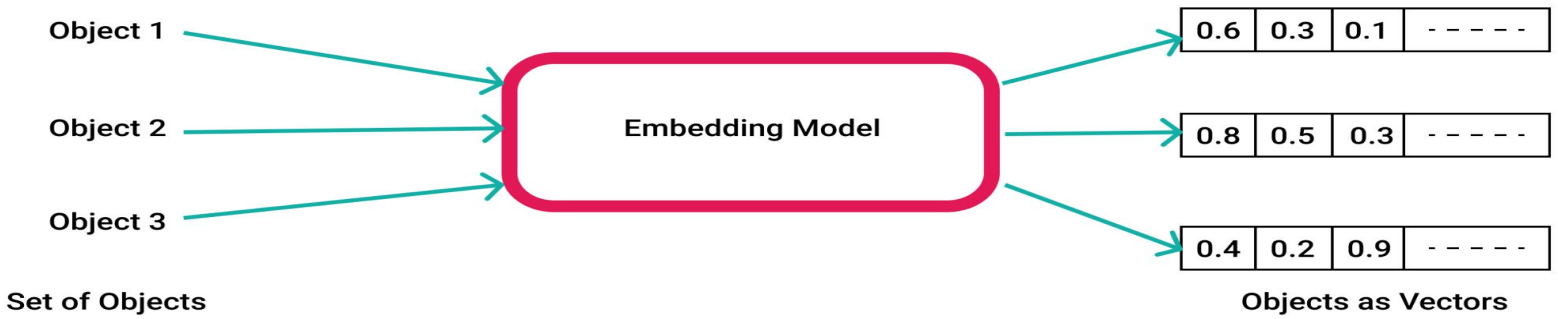
- Embedding or Word Vector is a numeric vector input that represents a word in a lower-dimensional space.
- The representation is a real-valued vector that encodes the meaning of the word in such a way that the words that are closer in the vector space are expected to be similar in meaning.
- Different embedding model have different vector size.

## Need for Embedding?

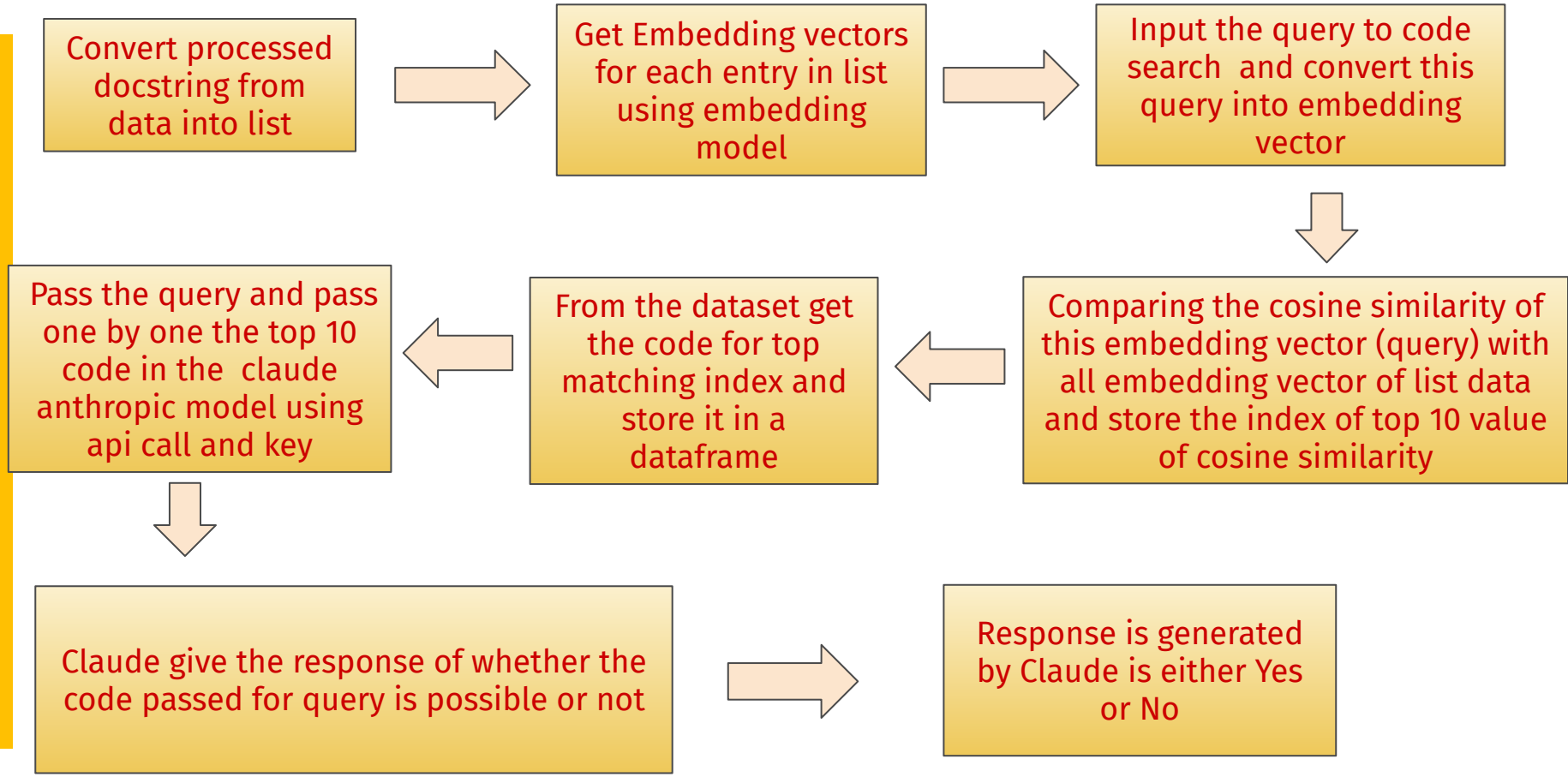
- To reduce dimensionality
- To use a word to predict the words around it.
- Inter-word semantics must be captured.

# How are Word Embeddings used?

- Take the words —-> Give their numeric representation —-> Use in training or inference.
- To represent or visualize any underlying patterns of usage in the corpus that is used to train them.



# Work Flow


Convert processed docstring from data into list
Get Embedding vectors for each entry in list using embedding model
Input the query to code search and convert this query into embedding vector
Comparing the cosine similarity of this embedding vector (query) with all embedding vector of list data and store the index of top 10 value of cosine similarity
From the dataset get the code for top matching index and store it in a dataframe
Pass the query and pass one by one the top 10 code in the claude anthropic model using api call and key
Claude give the response of whether the code passed for query is possible or not
Response is generated by Claude is either Yes or No

# Claude Anthropic model

- The query from user and one by one top 10 code are passed into Claude.
- Code is converted to dataframe as it is the data in this case.
- Human response is the query in this case.
- We have to the define the system prompt like what we want from the claude do with our data.
- Like In this we are matching the code for query is possible or not accordingly we define the system .

```
# Initialize the ChatAnthropic object
chat = ChatAnthropic(anthropic_api_key=key ,temperature=0, model_name="claude-3-haiku-20240307")

# Defining system message with task description and data
system = (
""" Your task is to provide a response of only 'YES' if there is a 75 percentage matching of code for
    the human input and the data,
    or only 'No' if there isn't,
    when comparing the data to human input. Do not generate unnecessary response give 'YES' or 'NO' only.

data: {data}
human: {human}
"""
)

# Creating the ChatPromptTemplate
prompt = ChatPromptTemplate.from_messages([("system", system), ("human",human)])

# Creating the chain combining prompt and chat
chain = prompt | chat

# Invoking the chain with data and human input
response=chain.invoke(
{
     "data": data,
    "human": human,
}
)
return response
```

# Example :

Input from user and get the top 10 code for this query and pass this query to the claude api.

```
questions ='binary search function with complexity log n'
```

```
top_match_code=get_top_10_code(questions,embeddings_multilingual_e5_large_instruct,model_2)
```

Response from the claude api after matching the query with top code

```
print(check_response(questions,top_match_code))
```

```
AIMessage(content='YES')
```

# Testing data:

- Testing data is generated by our own .
- Testing data is docstring and it is generated by rephrasing and changing some words in the docstring.
- Number of query is testing data is 57.
- The MAP(mean average precision) of testing data is measured on the basis of how many times the response from the claude is yes for all the 57 queries ,means  for each query how many times the code is correctly retrieved or not.
- Precision of model = (Total  number of Yes / size of query data ) *100

# Result

- For a particular Query , top most matched docstring and code are fetched from the dataset
- Checked whether this code can be possible for query or not.
- Match column shows code is possible or not.

| | Query | Docstrings | Code | Match |
|---|---|---|---|---|
| 137 | apply processing running comparison like chang... | change array name searching array renaming | def rename_scalar(self, old_name, new_name, pr... | YES |
| 372 | Resizes the image according to width and size | return squared resized image | def no_crop(im, min_sz=None, interpolation=cv2... | NO |
| 32 | Get the status of a process with a particular ... | write current process pid pidfile location | def write_pid(self, pid=None):\n """Write t... | YES |
| 38 | Get the status of a process with a particular ... | check process running name | def get_pid(name):\n """Check if process is... | NO |
| 311 | Get all repositories list from project | list package one configured repos | def _repo_packages(self, args, search=False):\... | YES |
| 207 | binary search function complexity log n | time complexity dfs v e space complexity v | def top_sort(graph):\n """ Time complexity ... | YES |
| 130 | apply processing running comparison like chang... | applies post processing running comparison cha... | def _post_process_output(res):\n """\n A... | NO |
| 48 | calculate the cosine similarity of the tf idf... | return cosine similarity binary vector one len... | def cosine_similarity_vec(num_tokens, num_remo... | YES |
| 490 | Check if a username/password combination is va... | check username password combination valid | def _check_auth(self, username: str, password:... | YES |
| 339 | Returns a vocabulary after eliminating the wor... | build word frequency list incoming string | def counter_from_str(self, string):\n "... | YES |

# Embeddings Model

**Model 1** : sentence-transformers/all-mpnet-base-v2

- The embedding vector size is 768.
- Token size is 514.
- MAP@10 (mean average precision) of model_1 : 66.49

**Model 2** : intfloat/multilingual-e5-large-instruct

- This model has 24 layers and the embedding vector size is 1024.
- Token size is 514.
- MAP@10 (mean average precision) of model_2 : 63.68

# Embeddings Model

**Model 3** : intfloat/e5-base-v2

- This model has 12 layers and the embedding vector size is 768.
- Token size is 512.
- MAP@10 (mean average precision) of model_3 : 65.614

**Model 4** : mixedbread-ai/mxbai-embed-2d-large-v1

- The embedding vector size is 1024.
- Token size is 512.
- MAP@10 (mean average precision) of model_4 : 65.614