Company Matching

Notebook Ownership: Superpowers
Contributors: Nina Ning

Directions

1. Upload your CSV file of company information from the database to the same directory as this notebook.
2. Change the variable `INPUT_CSV` in the cell below to the name of your CSV file.

Below are **optional** steps.

3. You may want to combine the results from this matching with a previous matching output. To do so, change the variable `UPDATE_CSV` to the name of your CSV file generated from this notebook. If you do not want to combine the results, you may set this variable to `None`.
4. You may want to change the name of your output CSV file. To do so, change the variable `OUTPUT_CSV`.
5. You may want to change the value of `MINIMUM_MATCH`, which defines the minimum match percentage between 2 companies to qualify them to be a top match for each other.
6. You may want to change the number of `MATCHES_TO_CONSIDER`. If there aren't enough companies that match, then this notebook falls back to the top `MATCHES_TO_CONSIDER` companies, regardless of how well they match. It is recommended to set this value between 3 and 5.
7. You may need to change the list of hubs. The cities are matched to states. To change a hub matched to a state, simply change the city associated. For example, change `'California': 'Sacramento'` to `'California': 'San Francisco'`.
8. Run the remaining cells without changing them.

Estimated Runtime: 1 minute per 100 rows

```
INPUT_CSV = "output_sampled_companies.csv"
UPDATE_CSV = "demo_companies_original.csv"
OUTPUT_CSV = "output.csv"
MINIMUM_MATCH = 0.45
MATCHES_TO_CONSIDER = 5
```

```
hubs = {'Alabama': 'Montgomery',
    'Alaska': 'Juneau',
    'Arizona':'Phoenix',
    'Arkansas':'Little Rock',
    'California': 'Sacramento',
    'Colorado':'Denver',
    'Connecticut':'Hartford',
    'Delaware':'Dover',
    'District of Columbia': 'District of Columbia',
    'Florida': 'Tallahassee',
    'Georgia': 'Atlanta',
    'Hawaii': 'Honolulu',
    'Idaho': 'Boise',
    'Illinois': 'Springfield',
    'Indiana': 'Indianapolis',
    'Iowa': 'Des Monies',
    'Kansas': 'Topeka',
    'Kentucky': 'Frankfort',
    'Louisiana': 'Baton Rouge',
    'Maine': 'Augusta',
    'Maryland': 'Annapolis',
    'Massachusetts': 'Boston',
    'Michigan': 'Lansing',
    'Minnesota': 'St. Paul',
    'Mississippi': 'Jackson',
    'Missouri': 'Jefferson City',
    'Montana': 'Helena',
    'Nebraska': 'Lincoln',
    'Nevada': 'Carson City',
    'New Hampshire': 'Concord',
    'New Jersey': 'Trenton',
    'New Mexico': 'Santa Fe',
    'New York': 'Albany',
    'North Carolina': 'Raleigh',
    'North Dakota': 'Bismarck',
    'Ohio': 'Columbus',
    'Oklahoma': 'Oklahoma City',
    'Oregon': 'Salem',
    'Pennsylvania': 'Harrisburg',
    'Rhode Island': 'Providence',
    'South Carolina': 'Columbia',
    'South Dakota': 'Pierre',
    'Tennessee': 'Nashville',
    'Texas': 'Austin',
    'Utah': 'Salt Lake City',
    'Vermont': 'Montpelier',
    'Virginia': 'Richmond',
    'Washington': 'Olympia',
    'West Virginia': 'Charleston',
    'Wisconsin': 'Madison',
    'Wyoming': 'Cheyenne'}
```

DO NOT CHANGE CELLS BELOW THIS COMMENT!

```
#Imports
import pandas as pd
import numpy as np
import json
import matplotlib.pyplot as plt

from math import radians
from tqdm.auto import tqdm
from geopy.geocoders import Nominatim
from geopy.extra.rate_limiter import RateLimiter
!pip install sentence-transformers
from sentence_transformers import SentenceTransformer
from sklearn.metrics.pairwise import cosine_similarity
from sklearn.metrics.pairwise import haversine_distances
```

```
Collecting sentence-transformers
  Downloading sentence_transformers-2.6.1-py3-none-any.whl (163 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 163.3/163.3 kB 1.7 MB/s eta 0:00:00
Requirement already satisfied: transformers<5.0.0,>=4.32.0 in /usr/local/lib/python3.10/dist-packages (from sentence-transformers) (4
```

```
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from sentence-transformers) (4.66.2)
Requirement already satisfied: torch>=1.11.0 in /usr/local/lib/python3.10/dist-packages (from sentence-transformers) (2.2.1+cu121)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from sentence-transformers) (1.25.2)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-packages (from sentence-transformers) (1.2.2)
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages (from sentence-transformers) (1.11.4)
Requirement already satisfied: huggingface-hub>=0.15.1 in /usr/local/lib/python3.10/dist-packages (from sentence-transformers) (0.20.3
Requirement already satisfied: Pillow in /usr/local/lib/python3.10/dist-packages (from sentence-transformers) (9.4.0)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.15.1->sentence-transformer
Requirement already satisfied: fsspec>=2023.5.0 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.15.1->sentence-tra
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.15.1->sentence-transformer
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.15.1->sentence-transfor
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.15.1->se
Requirement already satisfied: packaging>=20.9 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.15.1->sentence-trar
Requirement already satisfied: sympy in /usr/local/lib/python3.10/dist-packages (from torch>=1.11.0->sentence-transformers) (1.12)
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-packages (from torch>=1.11.0->sentence-transformers) (3.2.1)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from torch>=1.11.0->sentence-transformers) (3.1.3)
Collecting nvidia-cuda-nvrtc-cu12==12.1.105 (from torch>=1.11.0->sentence-transformers)
  Downloading nvidia_cuda_nvrtc_cu12-12.1.105-py3-none-manylinux1_x86_64.whl (23.7 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 23.7/23.7 MB 29.5 MB/s eta 0:00:00
Collecting nvidia-cuda-runtime-cu12==12.1.105 (from torch>=1.11.0->sentence-transformers)
  Downloading nvidia_cuda_runtime_cu12-12.1.105-py3-none-manylinux1_x86_64.whl (823 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 823.6/823.6 kB 53.4 MB/s eta 0:00:00
Collecting nvidia-cuda-cupti-cu12==12.1.105 (from torch>=1.11.0->sentence-transformers)
  Downloading nvidia_cuda_cupti_cu12-12.1.105-py3-none-manylinux1_x86_64.whl (14.1 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 14.1/14.1 MB 43.0 MB/s eta 0:00:00
Collecting nvidia-cudnn-cu12==8.9.2.26 (from torch>=1.11.0->sentence-transformers)
  Downloading nvidia_cudnn_cu12-8.9.2.26-py3-none-manylinux1_x86_64.whl (731.7 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 731.7/731.7 MB 1.3 MB/s eta 0:00:00
Collecting nvidia-cublas-cu12==12.1.3.1 (from torch>=1.11.0->sentence-transformers)
  Downloading nvidia_cublas_cu12-12.1.3.1-py3-none-manylinux1_x86_64.whl (410.6 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 410.6/410.6 MB 2.4 MB/s eta 0:00:00
Collecting nvidia-cufft-cu12==11.0.2.54 (from torch>=1.11.0->sentence-transformers)
  Downloading nvidia_cufft_cu12-11.0.2.54-py3-none-manylinux1_x86_64.whl (121.6 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 121.6/121.6 MB 5.5 MB/s eta 0:00:00
Collecting nvidia-curand-cu12==10.3.2.106 (from torch>=1.11.0->sentence-transformers)
  Downloading nvidia_curand_cu12-10.3.2.106-py3-none-manylinux1_x86_64.whl (56.5 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 56.5/56.5 MB 8.3 MB/s eta 0:00:00
Collecting nvidia-cusolver-cu12==11.4.5.107 (from torch>=1.11.0->sentence-transformers)
  Downloading nvidia_cusolver_cu12-11.4.5.107-py3-none-manylinux1_x86_64.whl (124.2 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 124.2/124.2 MB 4.6 MB/s eta 0:00:00
Collecting nvidia-cusparse-cu12==12.1.0.106 (from torch>=1.11.0->sentence-transformers)
  Downloading nvidia_cusparse_cu12-12.1.0.106-py3-none-manylinux1_x86_64.whl (196.0 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 196.0/196.0 MB 4.1 MB/s eta 0:00:00
Collecting nvidia-nccl-cu12==2.19.3 (from torch>=1.11.0->sentence-transformers)
  Downloading nvidia_nccl_cu12-2.19.3-py3-none-manylinux1_x86_64.whl (166.0 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 166.0/166.0 MB 5.4 MB/s eta 0:00:00
Collecting nvidia-nvtx-cu12==12.1.105 (from torch>=1.11.0->sentence-transformers)
  Downloading nvidia_nvtx_cu12-12.1.105-py3-none-manylinux1_x86_64.whl (99 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 99.1/99.1 kB 11.2 MB/s eta 0:00:00
Requirement already satisfied: triton==2.2.0 in /usr/local/lib/python3.10/dist-packages (from torch>=1.11.0->sentence-transformers) (2
Collecting nvidia-nvjitlink-cu12 (from nvidia-cusolver-cu12==11.4.5.107->torch>=1.11.0->sentence-transformers)
  Downloading nvidia_nvjitlink_cu12-12.4.127-py3-none-manylinux2014_x86_64.whl (21.1 MB)
```

TESTS

The following code checks for errors in the hyperparameters in the first cell. Common errors:

- `INPUT_CSV` is not the name of a CSV file
- `UPDATE_CSV` is not `None` or the name of a CSV file
- `MATCHES_TO_CONSIDER` is not an integer greater than or equal to 3
- `MINIMUM_MATCH` is not a number between 0 and 1

```
#Tests on Hyperparameters
assert(len(INPUT_CSV) > 4 and INPUT_CSV[-4:] == ".csv")
assert(UPDATE_CSV == None or (len(UPDATE_CSV) > 4 and UPDATE_CSV[-4:] == ".csv"))
assert(len(OUTPUT_CSV) > 4 and OUTPUT_CSV[-4:] == ".csv")
assert(type(MATCHES_TO_CONSIDER) == int and MATCHES_TO_CONSIDER >= 3)
assert(type(MINIMUM_MATCH) in (int, float) and MINIMUM_MATCH >= 0 and MINIMUM_MATCH <= 1)
```

```
#Read CSV
df = pd.read_csv(INPUT_CSV)
if UPDATE_CSV:
  update = pd.read_csv(UPDATE_CSV)
df.head(15)
```

	company_id	company_name	company_type	company_location	company_size	top_member
0	b81093c4-9ba1-466e-93ac-aa37d25d2e3b	Agape Family Worship Center	company	Rahway, New Jersey USA	-1	
1	dea2b265-4ac7-4be9-ba6e-2836059bb527	Quest Dental	company	Baltimore, Maryland USA	-1	
2	20a426b1-a073-42a2-b5a2-8ecc8fdf7f68	Britsch Consulting	company	Kirkland, Washington USA	-1	
3	8eda1cd7-9460-487e-a03e-664f76609b2d	Diane Gordon Catering	company	New York, New York USA	-1	
4	047acd46-8f9f-4775-8b2c-21b6e68125e2	Unibat	company	Santa Fe Springs, California USA	-1	
5	644a6c5d-1e4a-4f1c-8428-46ebfa739248	Pinnacle	company	Delray Beach, Florida USA	-1	
6	020076ed-8544-4838-9e60-4ebb60d14c4c	EliteFox	company	Longmont, Colorado USA	-1	
7	e5dfa1e5-ea7d-4822-a67e-4d302a121479	The Marriage Group	company	Port Huron, Michigan USA	-1	
8	6e59dbec-7a0e-4603-85ba-	Midwest Engineering &	company	Wadsworth, Ohio USA	-1	

TESTS

The following code checks for errors in the Dataframes read from the CSV files. Common Errors:

- `INPUT_CSV` does not contain a CSV generated directly from the database with all columns present
- `UPDATE_CSV` is not `None` and does not contain a CSV generated directly from the results of this notebook with all columns present

```
#Tests on DF
assert(
  set([
      "company_id", "company_name", "company_type", "company_location", "company_size",
      "top_members", "company_website", "company_description", "created"
  ]) == set(df.columns)
)
if UPDATE_CSV:
  assert(
    set([
        "company_id", "company_name", "company_description", "embeddings",
        "hub", "city", "state", "radians_lat_long", 'website','linkedin','twitter','facebook',"top1", "top2", "top3"
    ]) == set(update.columns)
  )
```

```
# Clean DF
df_change = df.copy()
df_change = df_change.astype('string')
df_change["company_website"] = df_change["company_website"].astype('object')
df_change["company_website"] = df_change["company_website"].apply(lambda x: x[1:-1].split(", "))
df_change[["website", "facebook", "twitter", "linkedin"]] = pd.DataFrame(
  df_change["company_website"].tolist(),
  index= df_change.index
)
df_change["city"] = df_change["company_location"].apply(lambda x: x.split(", ")[0])
df_change["state"] = df_change["company_location"].apply(lambda x: x[:-3].split(", ")[1])
df_change["country"] = df_change["company_location"].apply(lambda x: x[-3:])
df_change = df_change.drop(
  columns=[
    "company_location", "company_size", "top_members", "company_website", "created"
  ]
)
df_change = df_change[df_change['country'] == "USA"]
for site in ["website", "facebook", "twitter", "linkedin"]:
  df_change[site] = df_change[site].apply(lambda x: "" if x == "nan" else x.strip("'"))
df_change.index = np.arange(0,len(df_change))
if UPDATE_CSV:
    update = update[~update["company_id"].isin(df_change["company_id"])]


#View Data
df_change.head(15)
```

	company_id	company_name	company_type	company_description	
0	b81093c4-9ba1-466e-93ac-aa37d25d2e3b	Agape Family Worship Center	company	Agape Family Worship Center is a religious org...	https://
1	dea2b265-4ac7-4be9-ba6e-2836059bb527	Quest Dental	company	Quest Dental provides implant dentistry, pedia...	https://dentistr
2	20a426b1-a073-42a2-b5a2-8ecc8fdf7f68	Britsch Consulting	company	Britsch Consulting is a bookkeeping company sp...	https://britsch
3	8eda1cd7-9460-487e-a03e-664f76609b2d	Diane Gordon Catering	company	Diane Gordon Catering offers catering services...	https://www.dianegorc
4	047acd46-8f9f-4775-8b2c-21b6e68125e2	Unibat	company	Unibat is a manufacturer and distributor of el...	https://ww
5	644a6c5d-1e4a-4f1c-8428-46ebfa739248	Pinnacle	company	Pinnacle is a retirement plan administration f...	https://\
6	020076ed-8544-4838-9e60-4ebb60d14c4c	EliteFox	company	A top-tier liquidity provider in the United St...	https://
7	e5dfa1e5-ea7d-4822-a67e-4d302a121479	The Marriage Group	company	The Marriage Group engages in delivering onlin...	https://themai
8	6e59dbec-7a0e-4603-85ba-759054016a80	Midwest Engineering & Automation	company	Midwest Engineering & Automation is an automat...	https://site.midweste
9	ed617604-53de-c70d-cb84-e1a6da45e175	Carney Labs	company	Imagine a world where being human has no limits.	http:/
10	3ee899eb-d3d5-41b0-bed2-07b5517221e2	Paradigm Payroll Services	company	Paradigm Payroll Services is a consulting firm...	https://www.paradigmpay
11	cd8c789d-6958-4f0f-b89b-	JRJ Income Tax Service	company	JRJ Income Tax Service offers notary	

```
#View Data
update.head(15) if UPDATE_CSV else "No UPDATE_CSV"
```

	company_id	company_name	company_description	embeddings
0	26678c95-eaf4-455f-b51e-02d2b9f65c9c	Qualicel Global	Mobile School Bus Tracking - first of its kind	[-0.004780501127243042, -0.043737225234508514,...
1	9cfd2a4f-c64b-4b1b-ada6-23cffec6f420	RunningWorks Inc	RunningWorks is a non-profit running program f...	[-0.027341987937688828, -0.02274911478161812, ...
2	e6718aca-b01e-4cbf-9b1f-2fa6b97f138e	Columbia Western Machinery	Columbia Western Machinery is an equipment com...	[-0.025654803961515427, 0.053678687661886215, ...
3	a7070c66-067a-4683-8f35-e547ea7df4b5	Datatility	Datatility provides network bandwidth design, ...	[0.06466042250394821, -0.050697192549705505, 0...
4	84ad42c0-ab5b-4bfe-a045-1ad92d430d1a	Nixa Nursing & Rehab	Nixa Nursing & Rehab provides medical care, nu...	[-0.06526317447423935, -0.029920993372797966, ...
5	e6616c37-6648-403f-bde7-d87c0bf5b3d2	Alpha Omega Resources	Alpha Omega Resource is an environmental servi...	[-0.056730739772319794, -0.012195838615298271,...
6	80631d6c-b5d3-4fb0-9f49-aa12e2d96214	New England Cremation Supply	New England Cremation Supply is a supplier of ...	[-0.028002332895994186, -0.009229668416082859,...
7	677a76f2-bb71-4fc4-8705-0f4369d96e4b	SCC	SCC manufactures combustion controls products ...	[-0.028407808393239975, -0.03913872689008713, ...
8	bab6c48a-9649-4c4a-9c77-5c3f6a2b4915	Walton EMC	Walton EMC is a customer-owned power company.	[-0.015436659567058086, 0.043583858758211136, ...
9	3d1c8f3c-c807-46c4-86ea-cf625e332d74	Daikin America, Inc	Daikin America is a chemicals firm that manufa...	[-0.034601107239723206, -0.036990974098443985,...
10	cd004934-c3c0-4b30-8468-bdd57d32cac4	NorthEast Logistics Systems	NorthEast Logistics Systems offers GIS, consul...	[0.0019823797047138214, 0.012943543493747711, ...
11	87f8b8cd-7093-4d13-b1e4-	PDG Architects	PDG Architects is a full-service	[-0.02523866668343544, 0.0007488290430046618,

```
#Download Model
model = SentenceTransformer('sentence-transformers/all-roberta-large-v1')
```

```
/usr/local/lib/python3.10/dist-packages/huggingface_hub/utils/_token.py:88: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public model
  warnings.warn(
modules.json: 100%                                                  349/349 [00:00<00:00, 7.22kB/s]
config_sentence_transformers.json: 100%                             116/116 [00:00<00:00, 3.84kB/s]
README.md: 100%                                                     9.89k/9.89k [00:00<00:00, 532kB/s]
sentence_bert_config.json: 100%                                     53.0/53.0 [00:00<00:00, 2.12kB/s]
config.json: 100%                                                   650/650 [00:00<00:00, 34.2kB/s]
model.safetensors: 100%                                             1.42G/1.42G [00:10<00:00, 60.3MB/s]
tokenizer_config.json: 100%                                         328/328 [00:00<00:00, 17.2kB/s]
vocab.json: 100%                                                    798k/798k [00:00<00:00, 2.42MB/s]
merges.txt: 100%                                                    456k/456k [00:00<00:00, 1.87MB/s]
tokenizer.json: 100%                                                1.36M/1.36M [00:00<00:00, 4.13MB/s]
special_tokens_map.json: 100%                                       239/239 [00:00<00:00, 7.22kB/s]
1_Pooling/config.json: 100%                                         191/191 [00:00<00:00, 5.09kB/s]
```

```
#NLP
#Around 200 rows per minute
batch_size = 200
embeddings = []
for i in tqdm(range(0, len(df_change), batch_size)):
    sentences_batch = df_change["company_description"].iloc[i:i+batch_size].tolist()
    embeddings_batch = model.encode(sentences_batch, show_progress_bar=True)
    embeddings.append(embeddings_batch)
embeddings = np.concatenate(embeddings)
```

```
100%                                                 50/50 [51:40<00:00, 56.22s/it]
Batches: 100%                                                 7/7 [01:02<00:00, 6.66s/it]
Batches: 100%                                                 7/7 [01:02<00:00, 6.48s/it]
Batches: 100%                                                 7/7 [00:56<00:00, 6.12s/it]
Batches: 100%                                                 7/7 [01:04<00:00, 6.83s/it]
Batches: 100%                                                 7/7 [01:00<00:00, 6.37s/it]
Batches: 100%                                                 7/7 [01:02<00:00, 6.46s/it]
Batches: 100%                                                 7/7 [00:59<00:00, 6.26s/it]
Batches: 100%                                                 7/7 [01:08<00:00, 7.38s/it]
Batches: 100%                                                 7/7 [00:57<00:00, 6.17s/it]
Batches: 100%                                                 7/7 [01:01<00:00, 6.42s/it]
Batches: 100%                                                 7/7 [01:11<00:00, 7.15s/it]
Batches: 100%                                                 7/7 [01:11<00:00, 6.87s/it]
Batches: 100%                                                 7/7 [01:02<00:00, 6.87s/it]
Batches: 100%                                                 7/7 [01:01<00:00, 6.67s/it]
Batches: 100%                                                 7/7 [01:03<00:00, 6.50s/it]
Batches: 100%                                                 7/7 [01:04<00:00, 6.64s/it]
Batches: 100%                                                 7/7 [01:08<00:00, 7.25s/it]
Batches: 100%                                                 7/7 [01:06<00:00, 6.76s/it]
Batches: 100%                                                 7/7 [01:01<00:00, 6.44s/it]
Batches: 100%                                                 7/7 [01:05<00:00, 6.70s/it]
Batches: 100%                                                 7/7 [01:00<00:00, 6.39s/it]
Batches: 100%                                                 7/7 [01:01<00:00, 7.02s/it]
Batches: 100%                                                 7/7 [00:59<00:00, 6.61s/it]
Batches: 100%                                                 7/7 [00:59<00:00, 6.07s/it]
Batches: 100%                                                 7/7 [01:07<00:00, 6.60s/it]
Batches: 100%                                                 7/7 [01:01<00:00, 6.42s/it]
Batches: 100%                                                 7/7 [01:06<00:00, 7.80s/it]
Batches: 100%                                                 7/7 [01:00<00:00, 6.34s/it]
Batches: 100%                                                 7/7 [01:02<00:00, 6.77s/it]
Batches: 100%                                                 7/7 [01:00<00:00, 6.59s/it]
```

```
print(len(df_change['city']), len(df_change['state']))
```

```
9950 9950
Batches: 100%                                                 7/7 [01:02<00:00, 6.51s/it]
```

```
#Distance Difference
#Around 100 locations per minute
geolocator = Nominatim(user_agent="test", timeout=5)
locations = (df_change["city"] + ", " + df_change["state"]).tolist()
print(len(locations))
unique_locations = set(locations)
unique_lat_long = dict([(x, geolocator.geocode(x)[1]) for x in tqdm(unique_locations)])
lat_long = [unique_lat_long[x] for x in locations]
lat_long = [[radians(x[0]), radians(x[1])] for x in lat_long]
```

```
100%                                                 2796/2796 [23:35<00:00, 1.96it/s]
Batches: 100%                                                 7/7 [00:59<00:00, 6.06s/it]
```

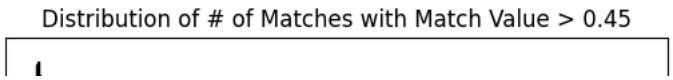
```
#Compute Similarities
print(len(lat_long), len(embeddings))
if UPDATE_CSV:
  update_embeddings = np.array([
    json.loads(x.replace("'", '"')) for x in update["embeddings"].tolist()
  ])
  update_lat_long = np.array([
    json.loads(x.replace("'", '"')) for x in update["radians_lat_long"].tolist()
  ])
  print(len(update_lat_long), len(update_embeddings))
  embeddings_similarity = cosine_similarity(
    np.vstack([embeddings, update_embeddings]) if update_embeddings.size else embeddings
  )
  distance = haversine_distances(np.vstack([lat_long, update_lat_long]) if update_lat_long.size else lat_long)
  total_similarity = embeddings_similarity
else:
  embeddings_similarity = cosine_similarity(embeddings)
  distance = haversine_distances(lat_long)
  total_similarity = embeddings_similarity

# Order Top X Matches by Distance
ordered_similarity = total_similarity.argsort()
bar = MINIMUM_MATCH * np.ones(len(total_similarity))
top_x = sum((total_similarity >= bar).T)
adjusted_top_x = list(map(lambda x: MATCHES_TO_CONSIDER + 1 if x < MATCHES_TO_CONSIDER + 1 else x, top_x))
top_similar = [list(ordered_similarity[i, -adjusted_top_x[i]:-1]) for i in range(len(ordered_similarity))]

print("Length of distance:", len(distance))
print("Length of top_similar:", len(top_similar))

# Check the length of distance and top_similar to ensure they match
assert len(distance) == len(top_similar), "Length mismatch between distance and top_similar"

# Sort top_similar based on distance
top_similar = [
    sorted(top_similar[a], key=lambda x: distance[a][x]) for a in range(0, len(top_similar))
]

max_size = max(map(len, top_similar))
padded_top_similar = np.array([x + [None] * (max_size - len(x)) for x in top_similar])
```

```
9950 9950
1995 1995
Length of distance: 11945
Length of top_similar: 11945
```

Start coding or generate with AI.

```
#Show Statistics
plt.hist(top_x-np.ones(len(top_x)), bins=np.linspace(-0.5, max_size-0.5, max_size+1), density=True, color="pink", edgecolor="black")
plt.title(f"Distribution of # of Matches with Match Value > {MINIMUM_MATCH}")
plt.xlabel(f"Number of Matches Above {MINIMUM_MATCH}")
plt.ylabel("Frequency")
```

```
Text(0, 0.5, 'Frequency')
```




```
#Add Embeddings and Radian Latitude/Longitude to the DF
df_change["embeddings"] = embeddings.tolist()
df_change["radians_lat_long"] = lat_long
if UPDATE_CSV:
  df_change = pd.concat([df_change, update], axis=0, ignore_index=True)

#Add Top Matches to the DF
df_change[["top1", "top2", "top3"]] = padded_top_similar[:,:3]
'''df_change[["top3", "top3_description", "top3_location"]] = np.array([
  df_change["top3"].apply(lambda x: df_change.loc[x, "company_name"]),
  df_change["top3"].apply(lambda x: df_change.loc[x, "company_description"]),
  df_change["top3"].apply(lambda x: df_change.loc[x, "city"] + ", " + df_change.loc[x, "state"])
]).T
df_change[["top2", "top2_description", "top2_location"]] = np.array([
  df_change["top2"].apply(lambda x: df_change.loc[x, "company_name"]),
  df_change["top2"].apply(lambda x: df_change.loc[x, "company_description"]),
  df_change["top2"].apply(lambda x: df_change.loc[x, "city"] + ", " + df_change.loc[x, "state"])
]).T
df_change[["top1", "top1_description", "top1_location"]] = np.array([
  df_change["top1"].apply(lambda x: df_change.loc[x, "company_name"]),
  df_change["top1"].apply(lambda x: df_change.loc[x, "company_description"]),
  df_change["top1"].apply(lambda x: df_change.loc[x, "city"] + ", " + df_change.loc[x, "state"])
]).T'''
df_change[["top1", "top2", "top3"]] = np.array([
    df_change["top1"].apply(lambda x: df_change.loc[x, "company_id"]),
    df_change["top2"].apply(lambda x: df_change.loc[x, "company_id"]),
    df_change["top3"].apply(lambda x: df_change.loc[x, "company_id"])
]).T
df_change["hub"] =  df_change["state"].apply(lambda x: hubs[x.strip()])

#Select for Columns and Save to CSV
df_save = df_change[[
  "company_id", "company_name", "company_description", "embeddings",
  "hub", "city", "state", "radians_lat_long",
  "website", "linkedin", "twitter", "facebook",
#  "top1", "top1_description", "top1_location",
#  "top2", "top2_description", "top2_location",
#  "top3", "top3_description", "top3_location"
  "top1", "top2", "top3"
]]
df_save.to_csv(OUTPUT_CSV, index=False)
df_save.head(15)
```

	company_id	company_name	company_description	embeddings	hub
0	b81093c4-9ba1-466e-93ac-aa37d25d2e3b	Agape Family Worship Center	Agape Family Worship Center is a religious org...	[-0.027864690870046616, -0.028021540492773056,...	Trenton
1	dea2b265-4ac7-4be9-ba6e-2836059bb527	Quest Dental	Quest Dental provides implant dentistry, pedia...	[-0.020875582471489906, -0.012126697227358818,...	Annapolis
	20a426b1-		Britsch Consulting is a		