In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
from skimpy import skim
```

In [40]:
```
df = pd.read_csv('/Users/cantu/Documents/Data Science Masters/Applied DS DSC68
df.head(25)
```

Out[40]:

	Entity	Code	Year	MMLU avg	Training computation (petaFLOP)	Organization
0	BLOOM	NaN	2022	39.13	412000000	HuggingFace, BigScience
1	BloombergGPT	NaN	2023	39.18	212000000	Bloomberg
2	Chinchilla	NaN	2022	67.50	588000000	Google DeepMind
3	GLM-130B	NaN	2022	44.80	312000000	Tsinghua KEG
4	GPT-2 (finetuned)	NaN	2019	32.40	36000	OpenAI
5	GPT-3 (davinci)	NaN	2020	43.90	393000000	OpenAI
6	GPT-3.5	NaN	2022	70.00	2580000000	OpenAI
7	GPT-4	NaN	2023	86.40	21000000000	OpenAI
8	GPT-NeoX-20B	NaN	2022	33.60	21200000	Eleuther
9	Gemini Ultra	NaN	2023	83.96	80000000000	Google DeepMind
10	Gopher (0.4B)	NaN	2021	25.70	751000	Google DeepMind
11	Gopher (1.4B)	NaN	2021	27.30	2520000	Google DeepMind
12	Gopher (280B)	NaN	2021	60.00	504000000	Google DeepMind
13	Gopher (7B)	NaN	2021	29.50	12800000	Google DeepMind
14	LLaMA (13B)	NaN	2023	46.90	78000000	Meta AI
15	LLaMA (33B)	NaN	2023	57.80	273000000	Meta AI
16	LLaMA (65B)	NaN	2023	63.40	548000000	Meta AI
17	LLaMA (7B)	NaN	2023	35.10	40200000	Meta AI
18	OPT	NaN	2022	35.99	172000000	Meta AI
19	PaLM (540B)	NaN	2022	69.30	2530000000	Google Research
20	PaLM (62B)	NaN	2022	53.70	296000000	Google Research
21	PaLM (62B+)	NaN	2022	62.80	493000000	Google Research
22	PaLM (8B)	NaN	2022	25.30	37400000	Google Research
23	PaLM-2	NaN	2023	78.30	8160000000	Google Research
24	code-davinci-002	NaN	2022	68.30	2580000000	OpenAI

In [3]:
```
skim(df)
```

```
/Users/cantu/anaconda3/lib/python3.11/site-packages/numpy/lib/histograms.py:88
3: RuntimeWarning: invalid value encountered in divide
  return n/db/n.sum(), bin_edges
```

```
┌─────────────────────────────────────────────────────── skimpy summary ──────
│          Data Summary                  Data Types
│ ┏━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━┓ ┏━━━━━━━━━━━━━┳━━━━━━━┓
│ ┃ dataframe          ┃ Values ┃ ┃ Column Type ┃ Count ┃
│ ┡━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━┩ ┡━━━━━━━━━━━━━╇━━━━━━━┩
│ │ Number of rows     │ 25     │ │ string      │ 2     │
│ │ Number of columns  │ 6      │ │ float64     │ 2     │
│ └────────────────────┴────────┘ │ int64       │ 2     │
│                                 └─────────────┴───────┘
│                                                    number
│ ┏━━━━━━━━━━━┳━━━━┳━━━━━━┳━━━━━━━━━━┳━━━━━━━━━━━┳━━━━━━━┳━━━━━━━━━━┳━━━━━
│ ┃ column_na ┃    ┃      ┃          ┃           ┃       ┃          ┃
│ ┃ me        ┃ NA ┃ NA % ┃ mean     ┃ sd        ┃ p0    ┃ p25      ┃ p50
│ ┡━━━━━━━━━━━╇━━━━╇━━━━━━╇━━━━━━━━━━╇━━━━━━━━━━━╇━━━━━━━╇━━━━━━━━━━╇━━━━━
│ │ Code      │ 25 │  100 │      nan │       nan │   nan │      nan │
│ │ Year      │  0 │    0 │     2022 │      1.02 │  2019 │     2022 │
│ │ MMLU avg  │  0 │    0 │    51.21 │     18.95 │  25.3 │     35.1 │
│ │ Training  │  0 │    0 │ 48500000 │ 162500000 │ 36000 │ 40200000 │ 3120
│ │ computati │    │      │       00 │        00 │       │          │
│ │ on        │    │      │          │           │       │          │
│ │ (petaFLOP │    │      │          │           │       │          │
│ │ )         │    │      │          │           │       │          │
│ └───────────┴────┴──────┴──────────┴───────────┴───────┴──────────┴─────
│                                                    string
│ ┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━┳━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━
│ ┃ column_name                         ┃ NA        ┃ NA %           ┃ words per row
│ ┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━╇━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━
│ │ Entity                              │         0 │              0 │
│ │ Organization                        │         0 │              0 │
│ └─────────────────────────────────────┴───────────┴────────────────┴──────────────
└──────────────────────────────────────────────────────────── End ──────────
```

Entity The name or code identifying the AI model or organization.

Year The timeframe during which the performance and knowledge tests were conducted.

MMLU avg The average performance metric, possibly denoting Mean Multi-Layered Understanding.

Training computation (petaFLOP) The amount of computational power utilized for training the AI model, measured in petaFLOPs.

Organization The entity responsible for developing or testing the AI model.

In [5]:
```
df2 = pd.read_csv('/Users/cantu/Documents/Data Science Masters/Applied DS DSC68
df2.head(10)
```

Out[5]:

	Entity	Code	Year	Design	Fabrication	Assembly, testing and packaging
0	China	CHN	2021	9.0	12.0	14.0
1	Italy	ITA	2021	NaN	2.0	NaN
2	Japan	JPN	2021	6.0	1.0	7.0
3	Malaysia	MYS	2021	NaN	NaN	2.0
4	Others	NaN	2021	9.0	NaN	5.0
5	Singapore	SGP	2021	NaN	NaN	2.0
6	South Korea	KOR	2021	6.0	11.0	13.0
7	Taiwan	TWN	2021	9.0	47.0	29.0
8	United States	USA	2021	61.0	27.0	28.0

In [6]:

```
skim(df2)
```

```
┌─────────────────────────────────────────────────────── skimpy summary ───────
│          Data Summary                    Data Types
│ ┏━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━┓  ┏━━━━━━━━━━━━━┳━━━━━━━┓
│ ┃ dataframe           ┃ Values ┃  ┃ Column Type ┃ Count ┃
│ ┡━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━┩  ┡━━━━━━━━━━━━━╇━━━━━━━┩
│ │ Number of rows      │ 9      │  │ float64     │ 3     │
│ │ Number of columns   │ 6      │  │ string      │ 2     │
│ └─────────────────────┴────────┘  │ int64       │ 1     │
│                                   └─────────────┴───────┘
│                                                    number
│ ┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━┳━━━━━━━━━┳━━━━━━━━━┳━━━━━━━━━┳━━━━━━━━┳━━
│ ┃ column_name                 ┃ NA  ┃ NA %    ┃ mean    ┃ sd      ┃ p0     ┃ p2
│ ┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━╇━━━━━━━━━╇━━━━━━━━━╇━━━━━━━━━╇━━━━━━━━╇━━
│ │ Year                        │   0 │       0 │    2021 │       0 │   2021 │ 2
│ │ Design                      │   3 │   33.33 │   16.67 │   21.77 │      6 │ 6
│ │ Fabrication                 │   3 │   33.33 │   16.67 │   17.56 │      1 │ 4
│ │ Assembly, testing and       │   1 │   11.11 │    12.5 │   10.84 │      2 │ 4
│ │ packaging                   │     │         │         │         │        │
│ └─────────────────────────────┴─────┴─────────┴─────────┴─────────┴────────┴──
│                                                    string
│ ┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━┳━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━
│ ┃ column_name                 ┃ NA      ┃ NA %           ┃ words per row
│ ┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━╇━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━
│ │ Entity                      │       0 │              0 │
│ │ Code                        │       1 │          11.11 │
│ └─────────────────────────────┴─────────┴────────────────┴─────────────────────
└──────────────────────────────────────────────────────────── End ─────────────
```

preprocessing

In [8]:

```
# remove ',' from column names and replace spaces with underscores
df.columns = df.columns.str.replace(',', '').str.replace(' ', '_')
df2.columns = df2.columns.str.replace(',', '').str.replace(' ', '_')
#df
#df2
```

In [9]:

```
df.drop(columns=['Code'], inplace=True)
df2.drop(columns=['Code'], inplace=True)
```

EDA

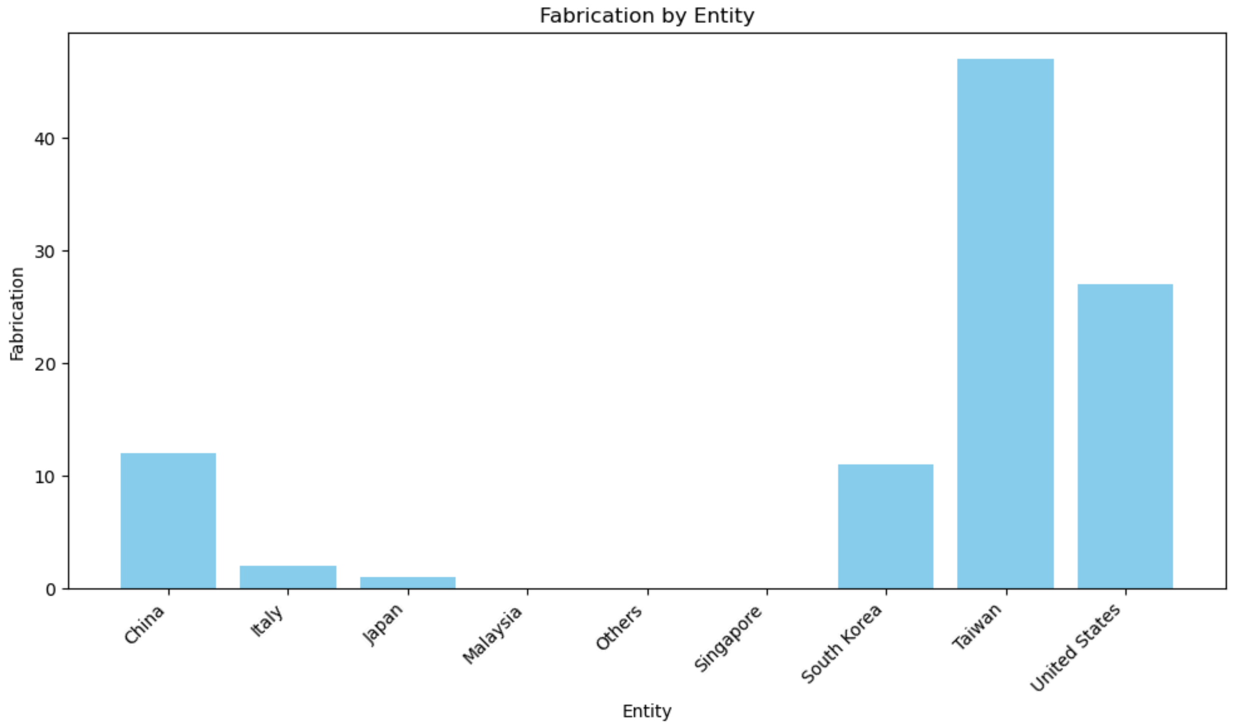
In [11]:

```
print(df2['Entity'])
```

```
0            China
1            Italy
2            Japan
3         Malaysia
4           Others
5        Singapore
6      South Korea
7           Taiwan
8    United States
Name: Entity, dtype: object
```

In [12]:

```
# Plotting
plt.figure(figsize=(10, 6))
plt.bar(df2['Entity'], df2['Fabrication'], color='skyblue')
plt.xlabel('Entity')
plt.ylabel('Fabrication')
plt.title('Fabrication by Entity')
plt.xticks(rotation=45, ha='right')  # Rotates x-axis labels for better visibi
plt.tight_layout()

# Display the plot
plt.show()
```

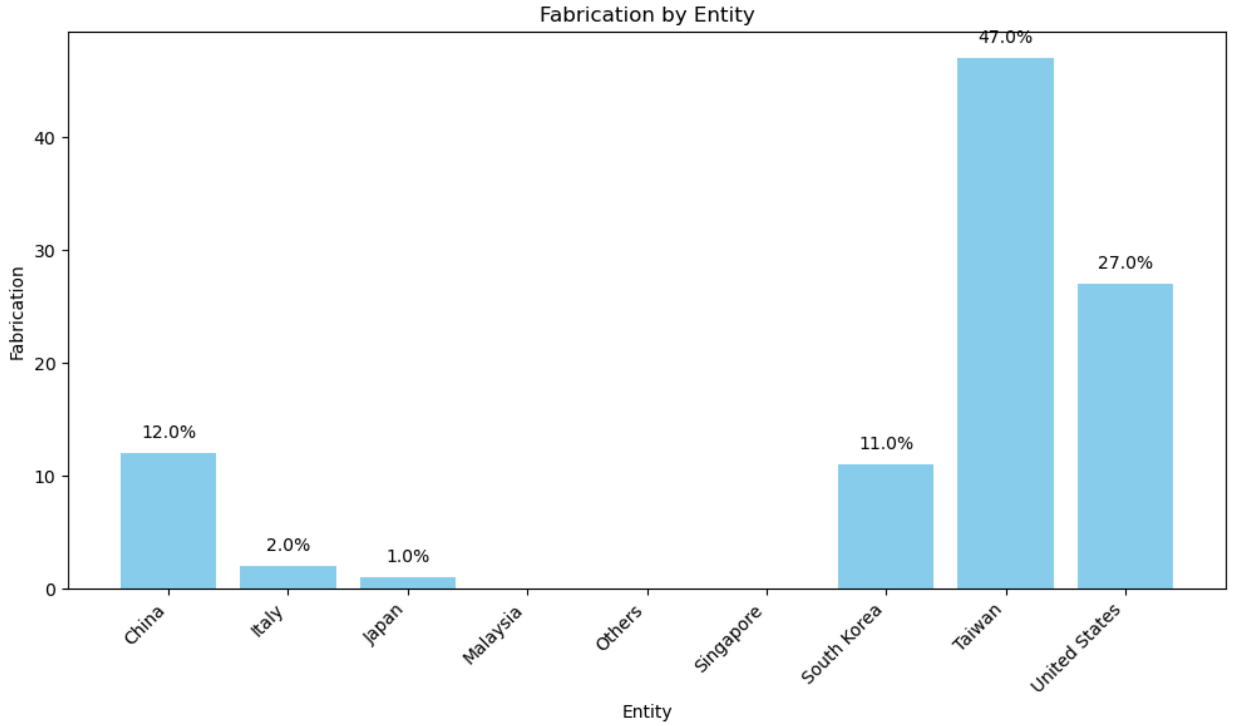


In [13]:

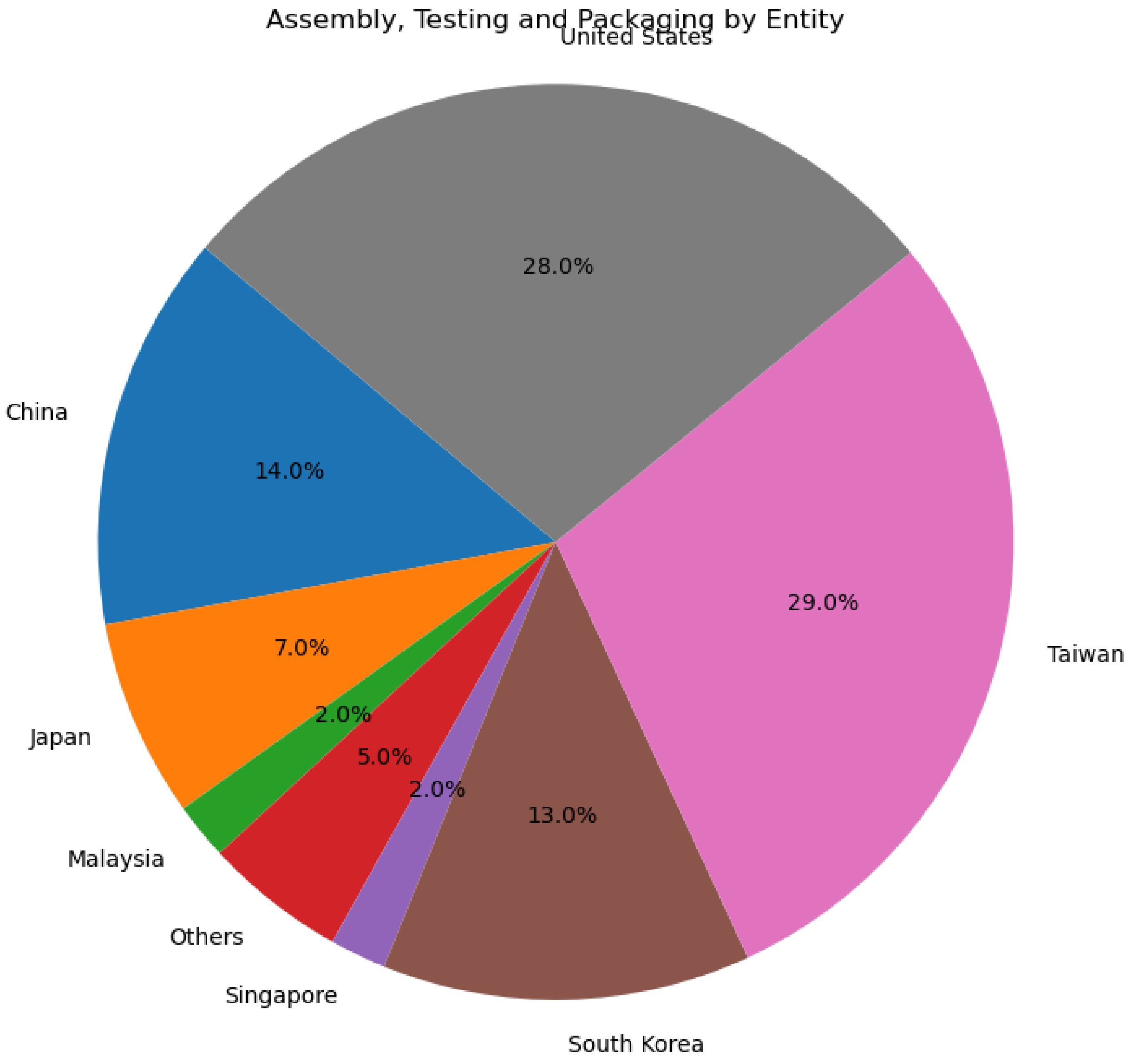
```
# Plotting
plt.figure(figsize=(10, 6))
bars = plt.bar(df2['Entity'], df2['Fabrication'], color='skyblue')
```

```
plt.xlabel('Entity')
plt.ylabel('Fabrication')
plt.title('Fabrication by Entity')
plt.xticks(rotation=45, ha='right')  # Rotates x-axis labels for better visibi
plt.tight_layout()

# Add percentage labels above the bars
for bar in bars:
    yval = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, yval + 1, f'{yval}%', ha='center

# Display the plot
plt.show()
```

```
posx and posy should be finite values
posx and posy should be finite values
posx and posy should be finite values
posx and posy should be finite values
posx and posy should be finite values
posx and posy should be finite values
```



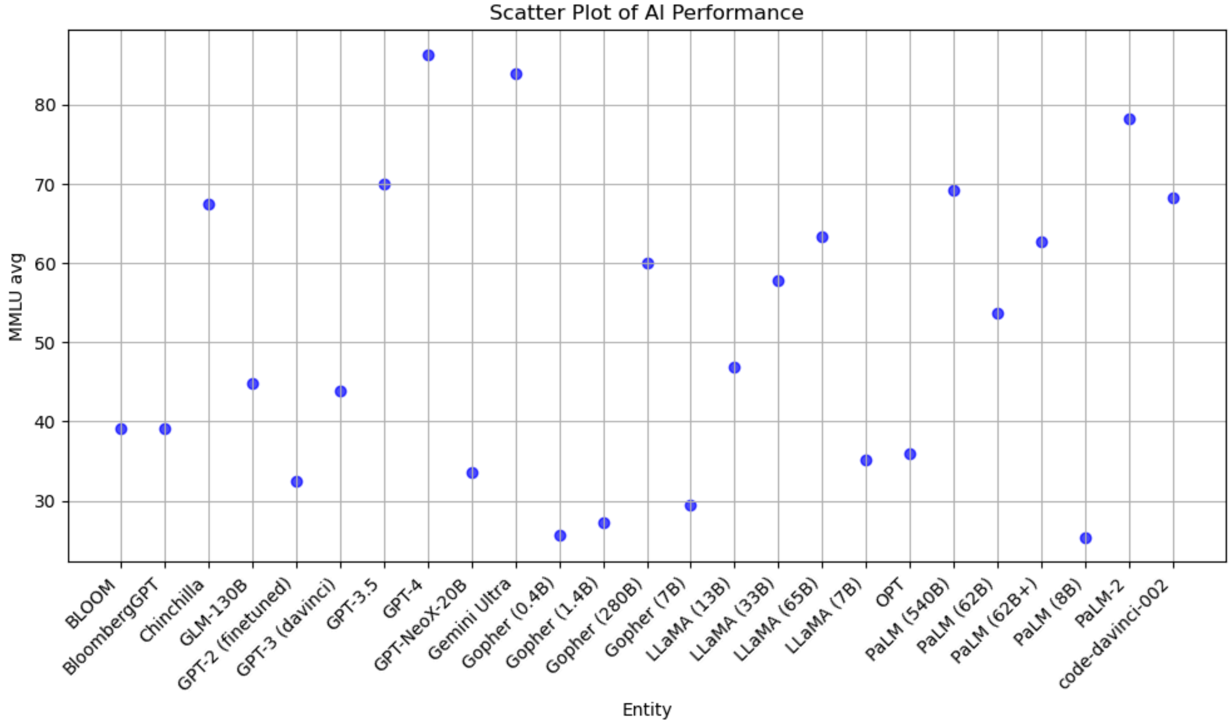
In [14]:
```
df2.dropna(subset=['Assembly_testing_and_packaging'], inplace=True)

# Plotting
plt.figure(figsize=(8, 8))
plt.pie(df2['Assembly_testing_and_packaging'], labels=df2['Entity'], autopct='
plt.title('Assembly, Testing and Packaging by Entity')
plt.axis('equal')  # equal aspect ratio ensures pie is drawn as a circle.

# Display the plot
plt.show()
```



In [15]:
```
print(df.columns)
```

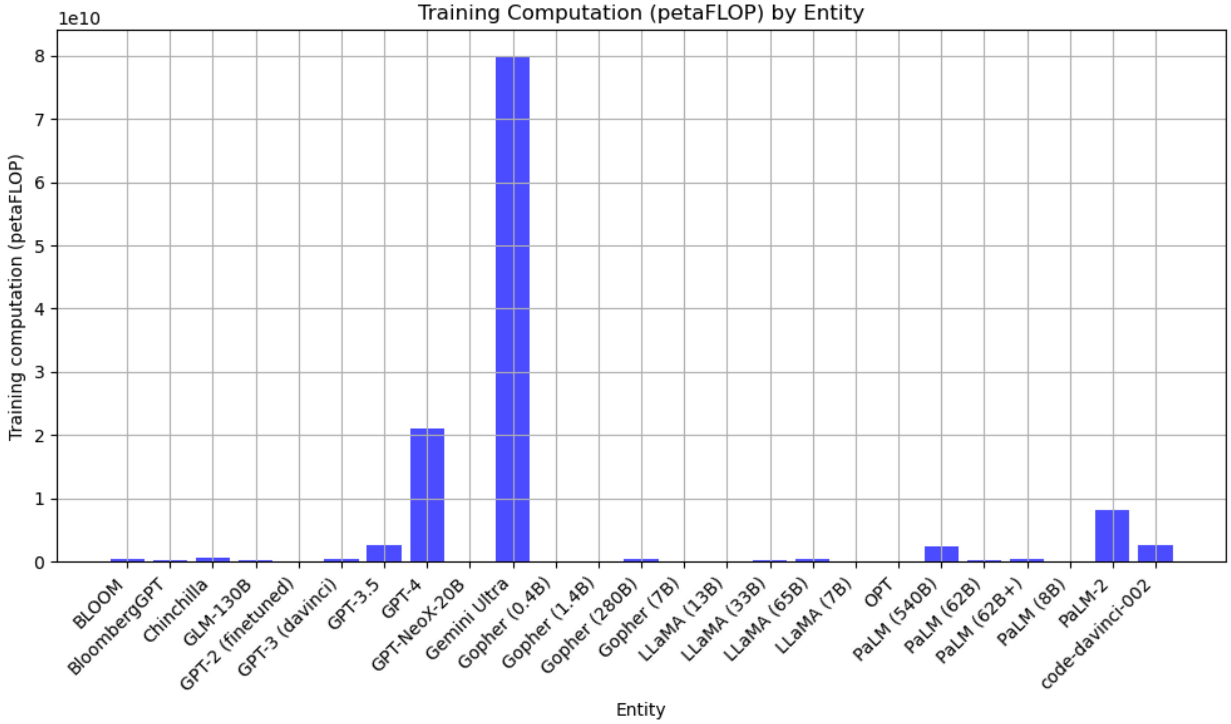
```
Index(['Entity', 'Year', 'MMLU_avg', 'Training_computation_(petaFLOP)',
       'Organization'],
      dtype='object')
```

In [16]:
```
# Plotting
plt.figure(figsize=(10, 6))
plt.scatter(df['Entity'], df['MMLU_avg'], color='blue', alpha=0.7)
plt.xlabel('Entity')
plt.ylabel('MMLU avg')
plt.title('Scatter Plot of AI Performance')
plt.xticks(rotation=45, ha='right')  # Rotates x-axis labels for better visibi
plt.grid(True)
plt.tight_layout()

# Display the plot
plt.show()
```



In [36]:

```
# Plotting
plt.figure(figsize=(10, 6))
plt.bar(df['Entity'], df['Training_computation_(petaFLOP)'], color='blue', alph
plt.xlabel('Entity')
plt.ylabel('Training computation (petaFLOP)')
plt.title('Training Computation (petaFLOP) by Entity')
plt.xticks(rotation=45, ha='right')  # Rotates x-axis labels for better visibi
plt.grid(True)
plt.tight_layout()

# Display the plot
plt.show()
```



```
In [ ]:
```

```
In [ ]:
```