In [ ]:
```
import requests
from bs4 import BeautifulSoup
```

Question 1

In [ ]:
```
def extract_text_from_wikipedia(url):
    response = requests.get(url)
    soup = BeautifulSoup(response.text, 'html.parser')
    paragraphs = soup.find_all('p')
    text = ' '.join([paragraph.get_text() for paragraph in paragraphs])
    return text

wikipedia_url = "https://en.wikipedia.org/wiki/Great_Hurricane_of_1780"

sentence = extract_text_from_wikipedia(wikipedia_url)
```

In [ ]:
```
# sentence = 'Life is short, eat dessert first'
```

In [ ]:
```
dc = {s:i for i,s in enumerate(sorted(sentence.replace(',', '').split()))}
print(dc)
```

{'"so': 0, '&': 1, "'Rock": 2, "'boiling": 3, "'divine": 4, "'injured'": 5, '("San': 6, '(150': 7, '(160': 8, '(295': 9, '(30': 10, '(320': 13, '(56': 14, '(7.6': 15, '(90': 16, '(San': 17, '(Santa': 18, '(except': 19, '(greater': 20, '(now': 21, '(the': 22, '...': 24, '1-minute': 25, '10': 29, '10.': 30, '100': 31, '10th]': 32, '11': 33, '11.[3]': 34, '114': 35, '11th;': 36, '12': 39, '12th': 40, '14': 42, '145': 43, '1492': 44, '15': 45, '16.[4]': 46, '1780': 47, '1780[2][1][3]': 48, '1851.[5]': 49, '1867': 50, '1899': 51, '18[3]': 52, '19': 53, '1928': 54, '1932': 55, '1953': 56, '1956)': 57, '1960': 58, '1960).[1]': 59, '1978': 60, '20': 61, '200': 64, '22': 65, '22000': 67, '240': 68, '25-foot': 69, '260': 70, '28500': 71, '40': 72, '4500': 73, '475': 74, '4:00': 75, '5': 77, '50': 78, '58': 79, '584': 80, '600': 81, '6000': 82, '6500': 83, '6:00': 84, '8:00': 85, '9.': 87, '90': 88, '9000': 90, 'A': 94, 'About': 95, 'According': 96, 'Admiral': 99, 'After': 100, 'Al': 101, 'Almighty': 102, 'American': 104, 'Americas': 105, 'Among': 106, 'An': 107, 'Andromeda': 108, 'Another': 109, 'Antigua': 110, 'Antilles': 111, 'At': 114, 'Atlantic': 125, 'Atlantic.': 126, 'August': 127, 'Austin': 128, 'Barbadoes': 129, 'Barbados': 134, 'Barbados.[3]': 135, 'Barbados.[9]': 136, 'Because': 138, 'Bermuda': 139, 'Bermuda.[3]': 140, 'Betsy': 141, 'Blanche': 142, 'British': 147, 'Brydges': 148, 'Bureau': 149, 'By': 150, 'Cabo': 151, 'Calixto': 152, 'Calixto"': 153, 'Callixtus': 155, 'Canada.': 156, 'Canada.[7]': 157, 'Cape': 159, 'Captaincy': 160, 'Caribbean': 162, 'Carlos': 163, 'Castle': 165, 'Castries': 167, 'Category': 168, 'Catholic': 169, 'Chancel': 170, 'Chapel': 171, 'Christ': 172, 'Christian': 173, 'Christopher': 174, 'Church': 175, "Church's": 176, 'Ciprian': 177, 'Ciriaco': 178, 'Clara': 179, 'Coming': 180, 'Cuban': 182, 'Deal': 183, 'Domingo.[3]': 184, 'Dominica': 185, 'Dominica.[3]': 186, 'Dominican': 187, 'Donna': 188, 'Duke': 189, 'Dutch': 191, 'Early': 192, 'East': 193, 'Endeavour': 194, 'English': 195, 'Estimates': 197, 'European': 198, 'Eustatia': 199, 'Eustatius': 202, 'Every': 203, 'Felipe': 204, 'Florida': 205, 'Florida).': 206, 'French': 209, 'General': 210, 'George': 211, "George's": 212, 'Grand': 213, 'Great': 216, 'Grenada': 217, 'Grenades': 218, 'Guadeloupe': 221, 'Guadeloupe.[3]': 222, 'HMS': 234, "Hall'": 235, 'Harper': 236, 'He': 239, 'Heavy': 240, 'Hemisphere.': 241, 'Here': 242, 'High': 243, 'Hispaniola': 244, 'However': 245, 'Hugh': 246, 'Hurricane': 252, 'I': 253, 'In': 255, 'Island;': 256, 'Islands.': 257, 'It': 259, 'Jamaica.': 260, 'Joshua': 261, 'José': 262, 'Junon.[13]:': 263, 'Kingstown.[3]': 264, 'Kitts': 267, 'Kitts.': 268, 'Lajas.': 269, 'Landsea': 270, 'Late': 271, 'Laurel': 272, 'Lesser': 273, 'Lord': 275, 'Lorenzo': 276, 'Lucia': 280, 'Lucia.': 281, "Lucy's": 282, 'Many': 283, 'Martinique': 289, 'Martinique.': 291, 'Meteorological': 292, "Michael's": 293, 'Millás': 294, 'Mitch': 295, 'Mona': 296, 'Most': 297, 'NOAA': 298, 'Narciso': 299, 'National': 300, 'New': 301, 'Newfoundland': 302, 'Northwesterly': 303, 'Ocean': 305, 'Oct.': 306, 'October': 325, 'October.[4]': 326, 'On': 329, 'Organization.': 330, 'Parker': 332, 'Passage': 333, 'Peter': 334, "Peter's": 335, "Philip's).": 336, 'Phoenix': 337, 'Pope': 338, 'Port': 341, 'Puerto': 347, 'Race': 348, 'Rear-Admiral': 349, 'Rector': 350, 'Republic': 351, 'Rev': 352, 'Revolution': 353, 'Revolutionary': 354, 'Rico': 358, 'Rico;': 360, 'Rodney': 362, "Rodney's": 363, 'Rojo': 364, 'Roman': 365, 'Roseau': 366, 'Rowley': 367, 'Rowley.': 368, 'Royal': 369, 'Saint': 378, 'Saint-Pierre': 380, 'Samaná.': 381, 'San': 386, 'Sandrik': 387, 'Sandwich': 388, 'Santo': 389, 'Scarborough': 390, 'Sea': 391, 'Sea.': 392, 'September': 393, 'Service)': 394, 'Seven': 395, 'Several': 396, 'Severe': 397, 'Since': 398, 'Sint': 401, 'Spanish).[1]': 402, 'Specifics': 403, 'St': 404, 'St.': 415, 'States': 416, 'Stirling': 417, 'Strong': 418, 'The': 442, 'This': 443, 'Thomas': 444, "Thomas's": 446, 'Thomas.': 447, 'Thunderer': 448, 'Tobago': 449, 'Turk': 450, 'United': 451, 'Verde': 452, 'Vice': 453, 'Victor': 454, 'Vincent': 457, 'W.': 458, 'War': 459, 'Weather': 461, 'Western': 462, 'When': 463, 'Wind': 464, 'Wm': 465, 'World': 466, 'York': 467, '[the': 468, 'a': 482, 'about': 490, 'affected': 491, 'affecting': 493, 'after': 500, 'afternoon': 501, 'ago': 502, 'all': 506, 'almost': 507, 'aloft': 508, 'alone': 509, 'also': 512, 'amazingly': 513, 'among': 514, 'an': 517, 'and': 567, 'any': 570, 'approach': 571, 'are': 574, 'area': 575, 'around': 576, 'arrival': 577, 'arrived': 578, 'as': 592, 'ashore.': 593, 'at': 601, 'attorney': 602, 'author': 60

3, 'away': 604, 'back': 606, 'backed': 607, 'bad': 608, 'bark': 609, 'basin': 611, 'basin.[3][4]': 612, 'bay': 613, 'be': 616, 'because': 617, 'been': 620, 'before': 622, 'began': 626, 'being': 628, 'blown': 630, 'boat': 631, 'both': 633, 'broke': 634, 'brought': 635, 'buildings': 636, 'buried': 637, 'but': 641, 'by': 649, 'can': 650, 'cannons': 651, 'carried': 652, 'caused': 656, 'causing': 659, 'chapel[]': 660, 'church': 664, 'churches': 666, "city's": 667, 'civilians': 668, 'closer': 669, 'closest': 670, 'coast': 673, 'coastline': 674, 'coastlines.': 675, 'come': 676, 'command': 677, 'completely': 678, 'conceive.': 679, 'considerable': 680, 'considered': 681, 'contesting': 682, 'continue': 683, 'continued': 684, 'control': 687, 'could': 690, 'crews': 691, 'curate': 692, 'cyclone': 695, 'cyclones': 696, 'damage': 702, 'damage.[3]': 703, 'damage:': 704, 'data': 705, 'database': 706, 'day': 710, 'days': 711, 'dead': 712, 'deadliest': 715, 'deadly': 716, 'deafening': 717, 'death': 720, 'deaths': 721, 'deaths.': 722, 'decades': 723, 'demolished': 724, 'demolition': 725, 'destroyed': 728, 'destroyed.': 729, 'destroying': 731, 'developed': 732, 'did': 735, 'died': 737, 'died.': 738, 'died;': 739, 'directions': 740, 'disappeared': 741, 'disastrous': 742, 'dismasted.[12][13]:': 743, 'down': 744, 'dramatic': 745, 'dreadful': 746, 'during': 749, 'early': 751, 'east': 753, 'eastern': 757, 'eight': 758, 'employees': 759, 'encountered': 760, 'entire': 761, 'escape': 762, 'estate': 763, 'estimate': 765, 'estimated': 768, 'even': 769, 'evening.': 770, 'ever': 772, 'exact': 773, 'example': 774, 'exceeding': 775, 'exceeds': 776, 'expanded': 777, 'extension': 778, 'eye': 780, 'fall': 781, 'family': 783, 'far': 784, 'feast': 786, 'feel': 787, 'feet': 788, 'fell': 789, 'female': 790, 'fierce': 791, 'figures': 792, 'first': 793, 'fleet': 799, 'flew': 800, 'for': 805, 'fort': 806, 'four': 807, 'frigate': 808, 'frigates': 810, 'from': 817, 'full': 818, 'fury': 819, 'gale': 820, 'gardens': 821, 'gender': 822, 'general': 823, 'goes': 824, 'going': 825, 'gradually': 827, 'great': 829, 'grounded': 830, 'had': 832, 'happened': 833, 'has': 836, 'have': 840, 'he': 842, 'hear': 843, 'heavy': 850, 'high': 853, 'higher': 855, 'his': 859, 'history': 860, 'history)': 861, 'history.': 862, 'hit': 863, 'hitting': 864, 'hospital': 865, 'hour': 866, 'hours': 867, 'house': 870, "house'": 871, 'houses': 875, 'however': 878, 'human': 879, 'hundred': 880, 'hurricane': 913, "hurricane's": 915, 'hurricane.': 916, 'hurricanes': 919, 'hurricanes.': 920, 'impact': 921, 'in': 957, 'including': 958, 'increased': 959, 'informal': 960, 'involved': 961, 'is': 965, 'island': 970, 'island.': 972, 'island.[3]': 973, 'islands': 974, 'islands.': 975, 'islands.[11]': 976, 'it': 988, 'it.': 990, 'its': 991, 'just': 992, 'killed.': 993, 'kilometers': 994, 'km': 998, 'km/h)': 999, 'km/h).[9]': 1000, 'km/h)[6]': 1001, 'known.': 1002, 'lack': 1003, 'laid': 1004, 'land.[10]': 1005, 'landfall': 1007, 'landfall[6]': 1008, 'largely': 1009, 'last': 1011, 'late': 1013, 'later': 1017, 'least': 1019, 'left': 1021, 'letter:': 1022, 'life.': 1023, 'lifted': 1024, 'likely': 1026, 'lives': 1027, 'loss': 1029, 'losses': 1031, 'lost': 1033, 'lost.': 1034, 'lot': 1035, 'm)': 1036, 'm).[3]': 1037, 'made': 1039, 'making': 1041, 'many': 1046, 'marginally': 1047, 'may': 1049, 'mention': 1050, 'mercy': 1051, 'meteorologist': 1052, 'mi)': 1057, 'midnight.': 1058, 'midst': 1059, 'miles': 1060, "military's": 1061, 'missed': 1062, 'moorings': 1063, 'more': 1065, 'morning': 1067, 'most': 1068, 'moving': 1069, 'mph': 1071, 'much': 1072, 'name': 1075, 'named': 1079, 'names': 1081, 'naming': 1083, 'near': 1087, 'neared': 1089, 'next': 1090, 'night': 1093, 'nineteen': 1094, 'none': 1095, 'noon': 1096, 'normal': 1097, 'north-northwest': 1098, 'north.': 1099, 'northeast': 1100, 'northeast.': 1101, 'northeastern': 1102, 'northwest': 1103, 'not': 1108, "o'clock": 1110, 'observed': 1112, 'occurred': 1113, 'occurring': 1114, 'of': 1174, 'off': 1179, 'official': 1181, 'officially': 1182, 'on': 1210, 'one': 1213, 'only': 1219, 'other': 1224, 'others': 1225, 'our': 1226, 'over': 1231, 'own': 1232, 'paralleled': 1233, 'parish': 1234, 'parsonage': 1235, 'part': 1236, 'passed': 1242, 'passing': 1244, 'past': 1245, 'path.': 1246, 'peaked': 1247, 'people': 1251, 'per': 1252, 'perhaps': 1253, 'perished': 1254, 'perished.': 1255, 'person': 1256, 'pleased': 1257, 'point': 1258, 'portion': 1262, 'possibility': 1263, 'possible': 1264, 'power.': 1265, 'precise.': 1266, 'present-day': 1268, 'previously': 1269, 'primarily': 1270, 'private': 1271, 'produced': 1273, 'province': 1276, 'rage': 1277, 'rain': 1278, 'ra

```
ins': 1279, 'range': 1280, 'rate': 1281, 'reached': 1282, 'record': 1283, 'recorde
d': 1287, 'recurved': 1288, 'region': 1289, 'relinquished': 1290, 'reported': 1293,
'returned': 1295, 'rough': 1296, 'ruined': 1297, 'said': 1298, 'saint': 1300, 'saint
s': 1301, 'sea': 1303, 'sea-officer': 1304, 'season': 1305, 'second-deadliest': 130
6, 'see': 1307, "services'": 1308, 'severe': 1310, 'severely': 1311, 'shelter': 131
2, 'ship': 1314, 'ships': 1323, 'short': 1324, 'show': 1325, 'single': 1326, 'situat
ion': 1327, 'sixth': 1328, 'slowly': 1329, 'so': 1332, 'soldiers': 1333, 'sought': 1
334, 'souls': 1335, 'south': 1336, 'southeast': 1339, 'southern': 1341, 'southwest':
1344, 'speed': 1346, 'speeds': 1347, 'standing': 1348, 'started': 1349, 'steadily':
1350, 'stopped': 1351, 'storm': 1360, 'storms': 1362, 'strength': 1363, 'strengthene
d': 1364, 'stripped': 1365, 'strong': 1367, 'struck': 1372, 'struck.[14]': 1373, 'st
ruts': 1374, 'subsequently': 1376, 'suffered': 1378, 'suggest': 1379, 'surge': 1382,
'sustained': 1383, 'system': 1384, 'than': 1389, 'thank': 1390, 'that': 1402, 'the':
1532, 'their': 1537, 'there': 1539, 'they': 1540, 'this': 1543, 'those': 1544, 'thou
ght.': 1545, 'thousands': 1546, 'through': 1550, 'throughout': 1551, 'tide': 1552,
'tides': 1553, 'till': 1554, 'time': 1555, 'to': 1581, 'toll': 1584, 'top': 1585, 't
otally': 1586, 'trace.': 1587, 'track': 1589, 'tracked': 1590, 'trees': 1591, 'tropi
cal': 1592, 'turned': 1596, 'two': 1601, 'ultimately': 1603, 'under': 1604, 'unknow
n': 1605, 'unknown.[8]': 1606, 'until': 1607, 'us': 1608, 'used': 1609, 'venerated':
1610, 'very': 1611, 'violence': 1612, 'violent': 1613, 'voices."[3]': 1614, 'wall':
1615, 'warehouses': 1616, 'was': 1638, 'washed': 1640, 'waste': 1641, 'waves': 1642,
'we': 1645, 'weakened': 1646, 'weakening': 1647, 'well': 1648, 'well.': 1649, 'wer
e': 1667, 'west-northwest': 1668, 'westerly': 1669, 'western': 1670, 'westward;': 16
71, 'when': 1673, 'where': 1674, 'which': 1681, 'while': 1684, 'wind': 1690, 'wind
s': 1698, 'with': 1708, 'without': 1710, 'woman': 1711, 'works': 1712, 'worst': 171
4, 'wrecked': 1717, 'wrecked.[3]': 1718, 'write': 1719, 'writes': 1720, 'writes:': 1
721, 'year': 1722, 'yes': 1724}
```

In [ ]:
```
import torch

sentence_int = torch.tensor([dc[s] for s in sentence.replace(',', '').split()])
print(sentence_int)
```

```
tensor([442, 216, 252,  ..., 393,  77,  59])
```

In [ ]:
```
torch.manual_seed(123)
embed = torch.nn.Embedding(len(sentence_int), 16)
embedded_sentence = embed(sentence_int).detach()

print(embedded_sentence)
print(embedded_sentence.shape)
```

```
tensor([[-0.8178,  0.8498, -0.1779,  ..., -0.6462,  1.0392, -0.6085],
        [ 0.0025, -0.1198, -0.3939,  ...,  1.3373,  0.2760,  1.1924],
        [ 0.3904,  0.0150, -0.6920,  ..., -1.3244,  1.3231,  1.4228],
        ...,
        [ 0.0434, -1.2346,  0.5360,  ...,  1.1410,  0.1475,  1.4444],
        [ 0.5383,  0.5548,  0.1676,  ...,  1.9944, -0.1130,  1.5312],
        [-0.5518,  0.8661,  0.1700,  ..., -0.6735, -0.5763, -0.9291]])
torch.Size([1725, 16])
```

In [ ]:
```
torch.manual_seed(123)

d = embedded_sentence.shape[1]

d_q, d_k, d_v = 24, 24, 28
```

```
W_query = torch.nn.Parameter(torch.rand(d_q, d))
W_key = torch.nn.Parameter(torch.rand(d_k, d))
W_value = torch.nn.Parameter(torch.rand(d_v, d))
```

In [ ]:
```
x_2 = embedded_sentence[1]
query_2 = W_query.matmul(x_2)
key_2 = W_key.matmul(x_2)
value_2 = W_value.matmul(x_2)

print(query_2.shape)
print(key_2.shape)
print(value_2.shape)
```

```
torch.Size([24])
torch.Size([24])
torch.Size([28])
```

In [ ]:
```
keys = W_key.matmul(embedded_sentence.T).T
values = W_value.matmul(embedded_sentence.T).T

print("keys.shape:", keys.shape)
print("keys:", keys)

print("values.shape:", values.shape)
print("values:", values)
```

```
keys.shape: torch.Size([1725, 24])
keys: tensor([[-1.0686, -3.1086, -0.8889,  ..., -0.0201, -0.9175,  0.4097],
        [-0.5862,  0.1007, -0.3218,  ..., -0.4393,  1.3399,  0.8820],
        [-0.6226, -0.2295,  0.7695,  ...,  2.0305, -0.8997,  1.5053],
        ...,
        [-0.8296,  0.5317, -1.7287,  ..., -1.0672,  1.2302, -1.1580],
        [ 1.7435,  3.0576,  2.6530,  ...,  1.6043,  3.1653,  2.7961],
        [ 1.5013,  3.5477,  0.2387,  ...,  0.6648,  1.3506,  1.4122]],
       grad_fn=<PermuteBackward0>)
values.shape: torch.Size([1725, 28])
values: tensor([[-0.4749, -0.5429, -0.6069,  ..., -0.4019,  1.1915,  0.2306],
        [ 2.8718,  0.7893,  0.9213,  ...,  0.8229,  2.8054,  0.1350],
        [ 1.9336, -0.0849, -2.0625,  ..., -0.0727, -1.2707, -2.0734],
        ...,
        [ 0.9851,  1.9797,  1.0178,  ..., -0.4025, -1.6145, -2.4468],
        [ 4.5607,  3.6346,  3.0015,  ...,  2.7838,  3.3667,  1.8448],
        [ 1.2109, -0.5297, -0.0923,  ...,  1.4707,  2.0475,  2.0686]],
       grad_fn=<PermuteBackward0>)
```

In [ ]:
```
omega_24 = query_2.dot(keys[4])
print(omega_24)
```

```
tensor(-0.7693, grad_fn=<DotBackward0>)
```

In [ ]:
```
omega_2 = query_2.matmul(keys.T)
print(omega_2)
```

```
tensor([-23.6476,  15.1263,  16.2770,  ...,   7.3063,  56.2660,   5.0327],
       grad_fn=<SqueezeBackward4>)
```

In [ ]:
```
import torch.nn.functional as F

attention_weights_2 = F.softmax(omega_2 / d_k**0.5, dim=0)
print(attention_weights_2)
```

```
tensor([3.1426e-12, 8.6019e-09, 1.0879e-08,  ..., 1.7432e-09, 3.8162e-05,
        1.0960e-09], grad_fn=<SoftmaxBackward0>)
```

In [ ]:
```
context_vector_2 = attention_weights_2.matmul(values)

print(context_vector_2.shape)
print(context_vector_2)
```

```
torch.Size([28])
tensor([6.4877, 6.5384, 5.5197, 4.8639, 4.4191, 9.7355, 3.7277, 4.6734, 4.7695,
        4.4612, 6.1600, 8.0053, 7.0177, 7.0056, 7.5777, 7.1089, 6.7831, 7.0839,
        4.7333, 6.4199, 6.5752, 7.4507, 6.6498, 5.4528, 5.7592, 5.6813, 5.4551,
        5.0441], grad_fn=<SqueezeBackward4>)
```

In [ ]:
```
h = 3
multihead_W_query = torch.nn.Parameter(torch.rand(h, d_q, d))
multihead_W_key = torch.nn.Parameter(torch.rand(h, d_k, d))
multihead_W_value = torch.nn.Parameter(torch.rand(h, d_v, d))
```

In [ ]:
```
multihead_query_2 = multihead_W_query.matmul(x_2)
print(multihead_query_2.shape)
```

```
torch.Size([3, 24])
```

In [ ]:
```
multihead_key_2 = multihead_W_key.matmul(x_2)
multihead_value_2 = multihead_W_value.matmul(x_2)
```

In [ ]:
```
stacked_inputs = embedded_sentence.T.repeat(3, 1, 1)
print(stacked_inputs.shape)
```

```
torch.Size([3, 16, 1725])
```

In [ ]:
```
multihead_keys = torch.bmm(multihead_W_key, stacked_inputs)
multihead_values = torch.bmm(multihead_W_value, stacked_inputs)
print("multihead_keys.shape:", multihead_keys.shape)
print("multihead_values.shape:", multihead_values.shape)
```

```
multihead_keys.shape: torch.Size([3, 24, 1725])
multihead_values.shape: torch.Size([3, 28, 1725])
```

In [ ]:
```
multihead_keys = multihead_keys.permute(0, 2, 1)
multihead_values = multihead_values.permute(0, 2, 1)
print("multihead_keys.shape:", multihead_keys.shape)
print("multihead_values.shape:", multihead_values.shape)
```

```
multihead_keys.shape: torch.Size([3, 1725, 24])
multihead_values.shape: torch.Size([3, 1725, 28])
```

Question 2

Tools used:

1. **SERP AI**: Wrapper around SerpAPI - a real-time API to access Google search results
2. **Calculator**: Perform calculations on response
3. **Buffer Memory**: Remembers previous conversational back and forths directly
4. **Chat OpenAI**: Wrapper around OpenAI large language models that use the Chat endpoint
5. **OpenAI Function Agent**: An agent that uses Function Calling to pick the tool and args to call

SERP AI and Calculator were fed into the `Allowed Tools` input of the OpenAI Function Agent, which allows it to use a simple calculator for simple math calculations or use SERP AI for complex searches. There aren't any specific attributes that can be customized in this configuration apart from the LLM used and the function agent.