Weight Sum of Helpfulness Metrics

Here we are going to create a function that prints out the importance of each helpfulness metric. Once we have the function, it can be used to look at the overall weights as well as weights for individual workers.

In [1]:
```
import pandas as pd
import numpy as np
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler

def calculate_metric_weights(df, worker_id=None):
    '''
    Calculate the weight of each helpfulness metric for a given writer.

    Input: csv file

    Output: dictionary of feature importance
    '''

    # Select relevant columns
    features = ['tone_similarity',      'pos_similarity',       'coherence_sco
    target = 'acceptance_status'

    # Filter for specific writer if worker_id is provided
    if worker_id:
        df = df[df['workerID'] == worker_id]

    # Drop rows with missing values
    df = df.dropna(subset=features + [target])

    # Convert target to binary, accepted is 1 and rejected is 0
    df[target] = df[target].apply(lambda x: 1 if x == 'accepted' else 0)

    if df[target].nunique() < 2:
        print(f"WorkerID: {worker_id} has only one class in target variable. Sk
        return {}

    # Standardize the features
    scaler = StandardScaler()
    X = scaler.fit_transform(df[features])
    y = df[target]

    # Train logistic regression model
    model = LogisticRegression()
    model.fit(X, y)

    # Get absolute values of coefficients and normalize them to sum to 1
    weights = np.abs(model.coef_[0])
    weights /= weights.sum()

    # Return dictionary of feature importance
    return dict(zip(features, weights))
```

In [2]:
```
df = pd.read_csv("all_metrics.csv")

# Get overall weights
weights = calculate_metric_weights(df)
print(weights)

# Get weights for all writers
unique_workers = df['workerID'].unique()

for worker in unique_workers:
  weights = calculate_metric_weights(df, worker_id=worker)
  print(f'WorkerID: {worker}, Weights: {weights}')
```

{'tone_similarity': 0.06601884014650725, 'pos_similarity': 0.3046119144345303, 'coherence_score': 0.39001169399995467, 'user_score': 0.08892239417966982, 'ai_score': 0.0759988759946788, 'ai_coverage': 0.0744362812446591}
WorkerID: A2WGW5Y3ZFBDEC, Weights: {'tone_similarity': 0.27247442149524514, 'pos_similarity': 0.20853556418609345, 'coherence_score': 0.01804222935152794, 'user_score': 0.22152220273780235, 'ai_score': 0.04337483926572697, 'ai_coverage': 0.2360507429636043}
WorkerID: AZCGF2D7QIO10, Weights: {'tone_similarity': 0.0843861667509652, 'pos_similarity': 0.23356246055293542, 'coherence_score': 0.2703854656448676, 'user_score': 0.02289716796762064, 'ai_score': 0.060811963911849315, 'ai_coverage': 0.32795677517176175}
WorkerID: A345TDMHP3DQ3G, Weights: {'tone_similarity': 0.07302832654081684, 'pos_similarity': 0.3281602311298298, 'coherence_score': 0.43771424037896545, 'user_score': 0.02728187352759628, 'ai_score': 0.10472128716957663, 'ai_coverage': 0.029094041253215016}
WorkerID: A2W121DQXNQK1, Weights: {'tone_similarity': 0.3347694846450531, 'pos_similarity': 0.049787541643277575, 'coherence_score': 0.42016876453371615, 'user_score': 0.047945452018514974, 'ai_score': 0.06821444631136954, 'ai_coverage': 0.07911431084806857}
WorkerID: A324VBRLXHG5IB, Weights: {'tone_similarity': 0.12264908091572031, 'pos_similarity': 0.24970869725434935, 'coherence_score': 0.14844117583048339, 'user_score': 0.09476588777326503, 'ai_score': 0.19793119559015931, 'ai_coverage': 0.1865039626360226}
WorkerID: A2ONILC0LZKG6Y, Weights: {'tone_similarity': 0.3619372945278562, 'pos_similarity': 0.005600430693059152, 'coherence_score': 0.315819163662672, 'user_score': 0.07643358869410546, 'ai_score': 0.0017114141138325804, 'ai_coverage': 0.23849810830847457}
WorkerID: A3OLRWACCCCUTU, Weights: {'tone_similarity': 0.08131223984405175, 'pos_similarity': 0.10364820160097976, 'coherence_score': 0.19581632152381673, 'user_score': 0.1302502907280743, 'ai_score': 0.2407343601832519, 'ai_coverage': 0.2482385861198256}
WorkerID: A2EED3HLTA96CP, Weights: {'tone_similarity': 0.11247831115442251, 'pos_similarity': 0.044942835918885315, 'coherence_score': 0.36320778768291834, 'user_score': 0.23219764267480042, 'ai_score': 0.18592583436078933, 'ai_coverage': 0.06124758820818416}
WorkerID: A2QKAA5YS0P4CI, Weights: {'tone_similarity': 0.21608290368456276, 'pos_similarity': 0.07971120610835707, 'coherence_score': 0.36003425262626326, 'user_score': 0.0991328456809884, 'ai_score': 0.0394440023363694, 'ai_coverage': 0.20559478956345917}
WorkerID: A3S67QA0SQVPUJ, Weights: {'tone_similarity': 0.4297967431062105, 'pos_similarity': 0.11370555629691084, 'coherence_score': 0.20163588036133834, 'user_score': 0.02189577791367658, 'ai_score': 0.09987930772169777, 'ai_coverage': 0.13308673460016598}
WorkerID: A1198W1SPF1R4, Weights: {'tone_similarity': 0.2513623283980478, 'pos_similarity': 0.20765129432981966, 'coherence_score': 0.23359055412813381, 'user_score': 0.037746297925939004, 'ai_score': 0.10208042829200902, 'ai_coverage': 0.1675690969260507}
WorkerID: ANCIB6B6EBBIJ, Weights: {'tone_similarity': 0.0825141079752729, 'pos_similarity': 0.15572231985434654, 'coherence_score': 0.4144941326582837, 'user_score': 0.03833218736474527, 'ai_score': 0.03527534956691112, 'ai_coverage': 0.2736619025804403}
WorkerID: AZZA3J049G7R5, Weights: {'tone_similarity': 0.12167035022308541, 'pos_similarity': 0.13739530897041471, 'coherence_score': 0.10168741313280957, 'user_score': 0.13608839657736985, 'ai_score': 0.28200637884179824, 'ai_coverage': 0.22115215225452217}
WorkerID: A8C3WNWRBWUXO, Weights: {'tone_similarity': 0.27878467974883453, 'pos_similarity': 0.005445469682553114, 'coherence_score': 0.3452847302617273, 'user_score': 0.07479682107793352, 'ai_score': 0.15895722606485224, 'ai_coverage': 0.1367310731640993}
WorkerID: A1PTH9KTRO06EG, Weights: {'tone_similarity': 0.16866824642417189, 'p

os_similarity': 0.10766002857170615, 'coherence_score': 0.08381699817614667,
'user_score': 0.043011749317671114, 'ai_score': 0.33220519770556595, 'ai_cover
age': 0.2646377798047384}
WorkerID: A2QX3YJXAAHHVV, Weights: {'tone_similarity': 0.02227075484373906, 'p
os_similarity': 0.15015711658746644, 'coherence_score': 0.12918141227189123,
'user_score': 0.33155577002255937, 'ai_score': 0.12265027900072926, 'ai_covera
ge': 0.2441846672736146}
WorkerID: A23KAJRDVCVGOE, Weights: {'tone_similarity': 0.2778924591444195, 'po
s_similarity': 0.2121371531582977, 'coherence_score': 0.09576697033791119, 'us
er_score': 0.17949926694178184, 'ai_score': 0.205178805803195, 'ai_coverage':
0.02952534461439472}
WorkerID: A1QUQ0TV9KVD4C, Weights: {'tone_similarity': 0.05307964889805268, 'p
os_similarity': 0.24506291029051264, 'coherence_score': 0.23389613461060835,
'user_score': 0.21400691811853742, 'ai_score': 0.16535171452813702, 'ai_covera
ge': 0.0886026735541518}
WorkerID: A1VZSFHTU51JP0, Weights: {'tone_similarity': 0.30129688532446236, 'p
os_similarity': 0.179901301471041, 'coherence_score': 0.296073399694418, 'user
_score': 0.1021139041906102, 'ai_score': 0.03298120075252518, 'ai_coverage':
0.08763330856694321}
WorkerID: A3MYPYBVHX7FQ2, Weights: {'tone_similarity': 0.027258979891016298,
'pos_similarity': 0.29787049862987947, 'coherence_score': 0.22714904234787286,
'user_score': 0.06282420618110068, 'ai_score': 0.329527787832423, 'ai_coverag
e': 0.055369485117707715}
WorkerID: ASVRLMDNQBUD9, Weights: {'tone_similarity': 0.049076729693943194, 'p
os_similarity': 0.3395854460908356, 'coherence_score': 0.05169122223306372, 'u
ser_score': 0.2138700850365801, 'ai_score': 0.34124563523658863, 'ai_coverag
e': 0.004530881708988755}
WorkerID: A2YTQDLACTLIBA, Weights: {'tone_similarity': 0.04364429274625334, 'p
os_similarity': 0.03463489337680766, 'coherence_score': 0.2962386288665496, 'u
ser_score': 0.03760599300770428, 'ai_score': 0.11345383474531609, 'ai_coverag
e': 0.4744223572573691}
WorkerID: A1TW2BZRRS874Z, Weights: {'tone_similarity': 0.056624227228597765,
'pos_similarity': 0.21767088544215737, 'coherence_score': 0.2835483748377027,
'user_score': 0.24415735432023902, 'ai_score': 0.1394650815019773, 'ai_coverag
e': 0.05853407666932586}
WorkerID: A2OVX9UW5WANQE, Weights: {'tone_similarity': 0.23999049358699762, 'p
os_similarity': 0.07354420229317549, 'coherence_score': 0.22185044859814124,
'user_score': 0.134257328712716, 'ai_score': 0.13998140867402925, 'ai_coverag
e': 0.19037611813494032}
WorkerID: A2OVOVZBJYUO, Weights: {'tone_similarity': 0.029170787169659998, 'po
s_similarity': 0.10175955798548617, 'coherence_score': 0.25545250094331656, 'u
ser_score': 0.2995339835975659, 'ai_score': 0.1349963150176865, 'ai_coverage':
0.17908685528628485}
WorkerID: A1FVXS8IM5QYO8, Weights: {'tone_similarity': 0.1762849471484, 'pos_s
imilarity': 0.2702829890864802, 'coherence_score': 0.29389614296737787, 'user_
score': 0.061878855431015956, 'ai_score': 0.08593407763060719, 'ai_coverage':
0.11172298773611872}
WorkerID: A3O5RKGH6VB19C, Weights: {'tone_similarity': 0.008038606028347882,
'pos_similarity': 0.465967591675968, 'coherence_score': 0.12246384798340748,
'user_score': 0.15729716823714437, 'ai_score': 0.10820519964360312, 'ai_covera
ge': 0.13802758643152913}
WorkerID: ABL2FXYMI00T6, Weights: {'tone_similarity': 0.07027924220264964, 'po
s_similarity': 0.01475398078691304, 'coherence_score': 0.04791732002645549, 'u
ser_score': 0.3034657578335189, 'ai_score': 0.4023468519560347, 'ai_coverage':
0.16123684719442827}
WorkerID: AZLZA0Q87TJZO, Weights: {'tone_similarity': 0.12283063799795194, 'po
s_similarity': 0.006106735770993987, 'coherence_score': 0.32335270354104245,
'user_score': 0.06585278210668724, 'ai_score': 0.24510586963797595, 'ai_covera
ge': 0.2367512709453485}
WorkerID: A3P9TM5PRYBH90, Weights: {'tone_similarity': 0.03918785361088174, 'p

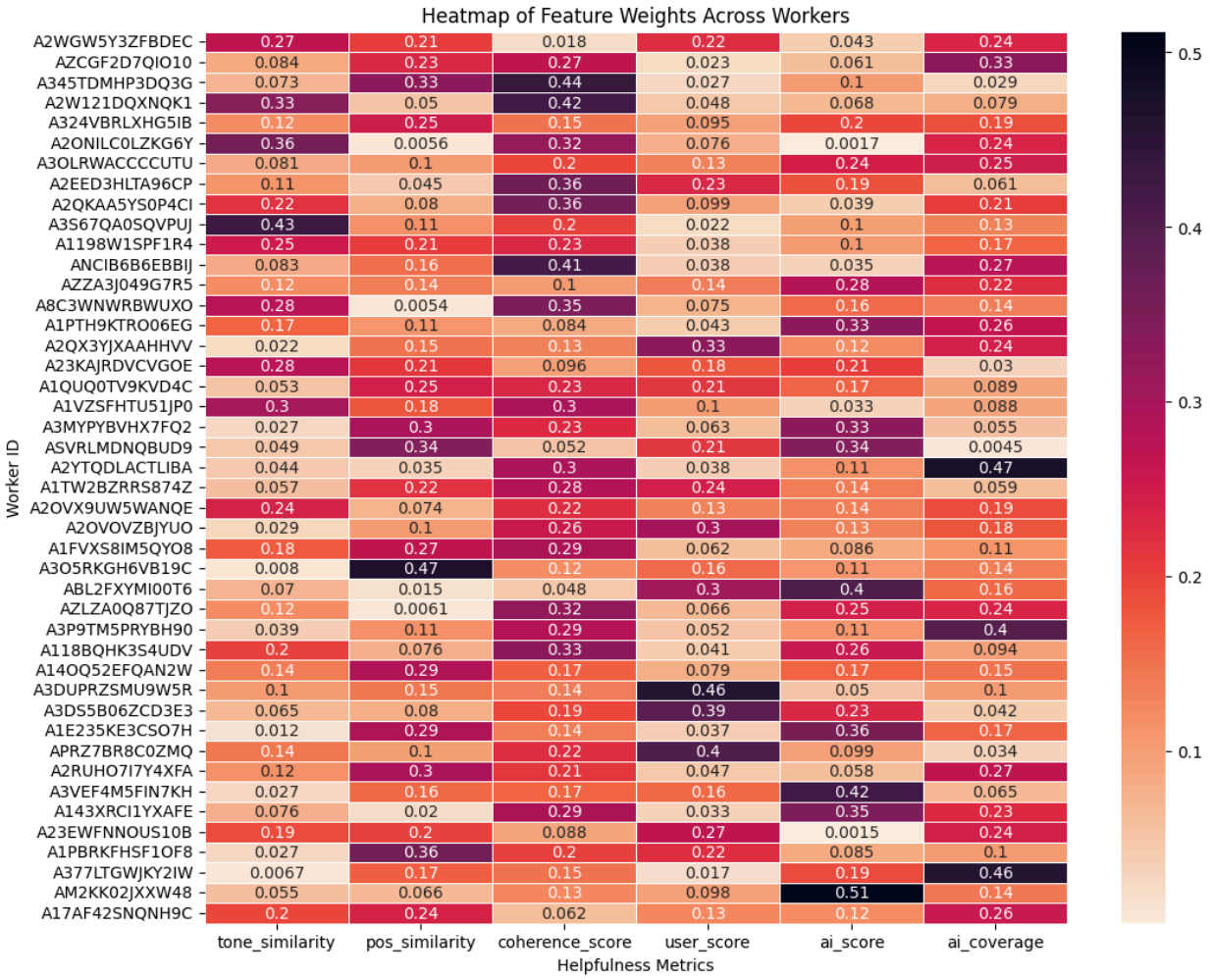
```
os_similarity': 0.11437187463238478, 'coherence_score': 0.2864065405479316, 'u
ser_score': 0.0517559105953281, 'ai_score': 0.11055741585081864, 'ai_coverag
e': 0.3977204047626552}
WorkerID: A118BQHK3S4UDV, Weights: {'tone_similarity': 0.2002110573139547, 'po
s_similarity': 0.07570781295065429, 'coherence_score': 0.32996743982937654, 'u
ser_score': 0.0411631221952255, 'ai_score': 0.2590181256450471, 'ai_coverage':
0.09393244206574189}
WorkerID: A17Q4QN6UE0EZC has only one class in target variable. Skipping...
WorkerID: A17Q4QN6UE0EZC, Weights: {}
WorkerID: AFIK3VBMMX6G6 has only one class in target variable. Skipping...
WorkerID: AFIK3VBMMX6G6, Weights: {}
WorkerID: A14OQ52EFQAN2W, Weights: {'tone_similarity': 0.14099115875550097, 'p
os_similarity': 0.28881842116703454, 'coherence_score': 0.17277442095548426,
'user_score': 0.079377297476725, 'ai_score': 0.16706490533897167, 'ai_coverag
e': 0.15097379630628346}
WorkerID: A1WKF2VH7TV0H2 has only one class in target variable. Skipping...
WorkerID: A1WKF2VH7TV0H2, Weights: {}
WorkerID: A3DUPRZSMU9W5R, Weights: {'tone_similarity': 0.10306128814722373, 'p
os_similarity': 0.14676953915452076, 'coherence_score': 0.13685466388659562,
'user_score': 0.46157505676688787, 'ai_score': 0.04960137578586202, 'ai_covera
ge': 0.10213807625891007}
WorkerID: A3DS5B06ZCD3E3, Weights: {'tone_similarity': 0.06507685443071755, 'p
os_similarity': 0.0800871044902143, 'coherence_score': 0.19312573637969094, 'u
ser_score': 0.3866174843546241, 'ai_score': 0.23340796489542173, 'ai_coverag
e': 0.04168485544933127}
WorkerID: A1E235KE3CSO7H, Weights: {'tone_similarity': 0.011926340899306788,
'pos_similarity': 0.2922323768098635, 'coherence_score': 0.13780166231198845,
'user_score': 0.03747212510373276, 'ai_score': 0.3554947943643866, 'ai_coverag
e': 0.16507270051072184}
WorkerID: APRZ7BR8C0ZMQ, Weights: {'tone_similarity': 0.14422406890653547, 'po
s_similarity': 0.09969714170438669, 'coherence_score': 0.22306549925012273, 'u
ser_score': 0.3998541839219472, 'ai_score': 0.09883894665448084, 'ai_coverag
e': 0.03432015956252716}
WorkerID: A2RUHO7I7Y4XFA, Weights: {'tone_similarity': 0.11675255188119175, 'p
os_similarity': 0.29504767336494647, 'coherence_score': 0.21206811678280463,
'user_score': 0.04706755960872501, 'ai_score': 0.05804692307326258, 'ai_covera
ge': 0.2710171752890695}
WorkerID: A3VEF4M5FIN7KH, Weights: {'tone_similarity': 0.026898121516466356,
'pos_similarity': 0.15954411185141434, 'coherence_score': 0.16833606082715677,
'user_score': 0.15909161842571454, 'ai_score': 0.4207948402982524, 'ai_coverag
e': 0.06533524708099557}
WorkerID: A173MXK429XAZQ has only one class in target variable. Skipping...
WorkerID: A173MXK429XAZQ, Weights: {}
WorkerID: A143XRCI1YXAFE, Weights: {'tone_similarity': 0.07559629694652995, 'p
os_similarity': 0.01984106482996159, 'coherence_score': 0.2937671575511489, 'u
ser_score': 0.03279793794079311, 'ai_score': 0.3493812251764123, 'ai_coverag
e': 0.22861631755515402}
WorkerID: A6KOTWP7N7RLU has only one class in target variable. Skipping...
WorkerID: A6KOTWP7N7RLU, Weights: {}
WorkerID: A23EWFNNOUS10B, Weights: {'tone_similarity': 0.19244809766410514, 'p
os_similarity': 0.20274286991185217, 'coherence_score': 0.08798776006304565,
'user_score': 0.27206977769915447, 'ai_score': 0.0015291821984571548, 'ai_cove
rage': 0.2432223124633855}
WorkerID: A1PBRKFHSF1OF8, Weights: {'tone_similarity': 0.026503305726128065,
'pos_similarity': 0.35634896144413136, 'coherence_score': 0.2027822708998881,
'user_score': 0.22485986468856065, 'ai_score': 0.08517165785034081, 'ai_covera
ge': 0.10433393939095109}
WorkerID: A377LTGWJKY2IW, Weights: {'tone_similarity': 0.006672610833185649,
'pos_similarity': 0.17075991838167662, 'coherence_score': 0.14935105917033986,
'user_score': 0.017109212161679956, 'ai_score': 0.19285537589588056, 'ai_cover
```

```
age': 0.4632518235572374}
WorkerID: A3HE29W5IDR394 has only one class in target variable. Skipping...
WorkerID: A3HE29W5IDR394, Weights: {}
WorkerID: AM2KK02JXXW48, Weights: {'tone_similarity': 0.05549902872318439, 'pos_similarity': 0.06572115507224002, 'coherence_score': 0.12609725754808485, 'user_score': 0.09806894960865416, 'ai_score': 0.5117993692493543, 'ai_coverage': 0.1428142397984822}
WorkerID: A17AF42SNQNH9C, Weights: {'tone_similarity': 0.19566843498875555, 'pos_similarity': 0.23901096553516982, 'coherence_score': 0.06176511385476217, 'user_score': 0.12640419199160735, 'ai_score': 0.11972732538282227, 'ai_coverage': 0.25742396824688296}
WorkerID: A394JO4NEPCY3M has only one class in target variable. Skipping...
WorkerID: A394JO4NEPCY3M, Weights: {}
```

In [3]:
```python
def calculate_weights_for_all_workers(df, unique_workers):
  '''
  Calculate the weight of each helpfulness metric for all writers.

  Input: csv file

  Output: dictionary of dictionary of feature importance
  '''

  worker_weights = {}
  for worker in unique_workers:
      weights = calculate_metric_weights(df, worker_id=worker)
      worker_weights[worker] = weights

  return worker_weights
```

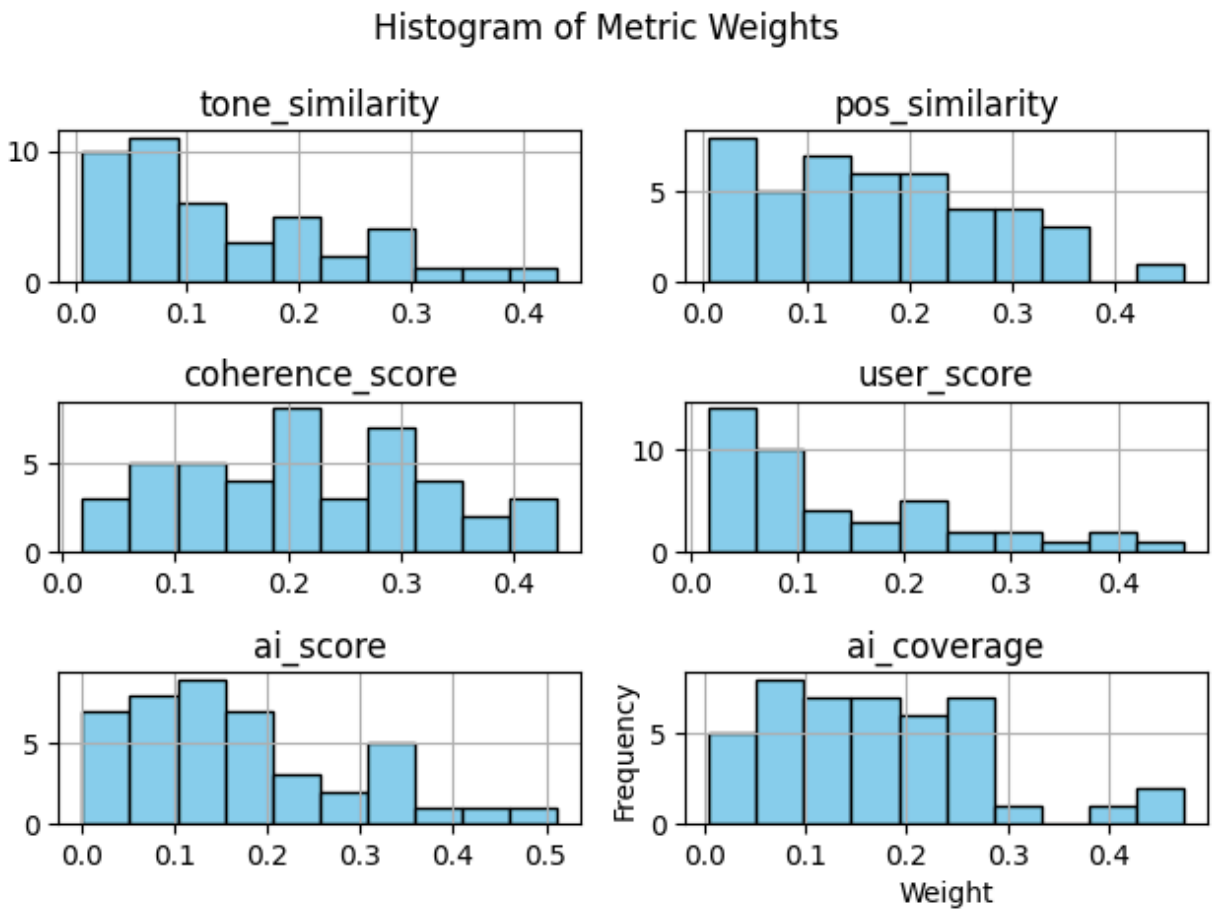
In [4]:
```python
import seaborn as sns
import matplotlib.pyplot as plt

worker_weights = calculate_weights_for_all_workers(df, unique_workers)
df_weights = pd.DataFrame.from_dict(worker_weights, orient='index')

plt.figure(figsize=(12, 10))
sns.heatmap(df_weights, annot=True, cmap="rocket_r", linewidths=0.5)
plt.xlabel("Helpfulness Metrics")
plt.ylabel("Worker ID")
plt.title("Heatmap of Feature Weights Across Workers")

plt.savefig('heatmap.png', dpi=300)
plt.show()
```

```
WorkerID: A17Q4QN6UE0EZC has only one class in target variable. Skipping...
WorkerID: AFIK3VBMMX6G6 has only one class in target variable. Skipping...
WorkerID: A1WKF2VH7TV0H2 has only one class in target variable. Skipping...
WorkerID: A173MXK429XAZQ has only one class in target variable. Skipping...
WorkerID: A6KOTWP7N7RLU has only one class in target variable. Skipping...
WorkerID: A3HE29W5IDR394 has only one class in target variable. Skipping...
WorkerID: A394JO4NEPCY3M has only one class in target variable. Skipping...
```

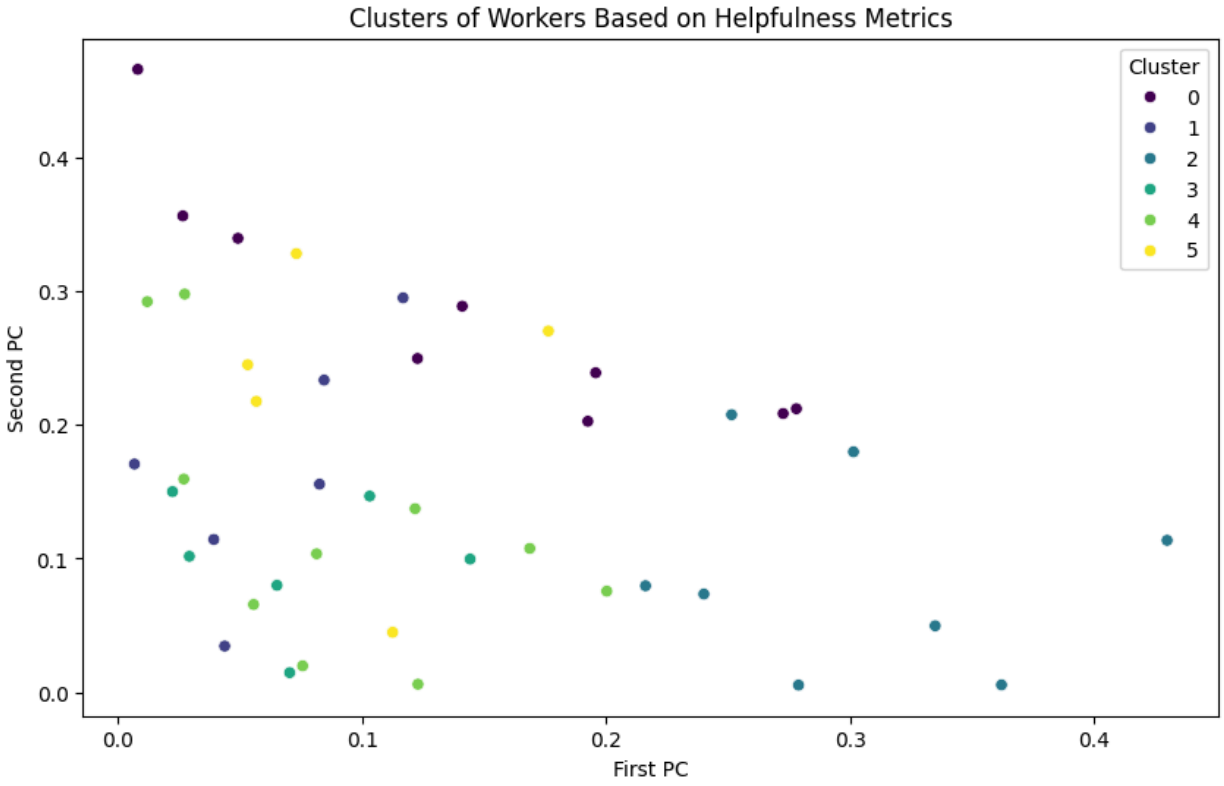


In [5]:
```
# histogram of metric weights
plt.figure(figsize=(4, 10))
df_weights.hist(bins=10, color='skyblue', edgecolor='black')
plt.suptitle('Histogram of Metric Weights', fontsize=12)
plt.xlabel('Weight', fontsize=10)
plt.ylabel('Frequency', fontsize=10)

#save img
plt.savefig('metric_weights_histogram.png')

plt.tight_layout()
plt.show()
```

```
<Figure size 400x1000 with 0 Axes>
```



In [6]:
```
from sklearn.cluster import KMeans
# Apply KMeans clustering
kmeans = KMeans(n_clusters=6, random_state=42)
df_weights['Cluster'] = kmeans.fit_predict(df_weights)

plt.figure(figsize=(10, 6))
sns.scatterplot(x=df_weights.iloc[:, 0], y=df_weights.iloc[:, 1], hue=df_weigh
plt.xlabel("First PC")
plt.ylabel("Second PC")
plt.title("Clusters of Workers Based on Helpfulness Metrics")
plt.legend(title="Cluster")
plt.show()
```


Clusters of Workers Based on Helpfulness Metrics
Cluster
0
1
2
3
4
5
Second PC
First PC
0.0
0.1
0.2
0.3
0.4