posthink_master

repository found here

This project marks the end of my forskerlinje (FL) year and the beginning of my master thesis. The data is retrived from a larger dataset from my FL-project. It consists of the answers to an open-end question participants provided after they were given an instruction to think as positively as possible when faced with an upcoming task. The question read: "When faced with a challenge, how are you supposed to think?" Our initial goal with this question was to check that participants acutally read the instruction, but as participants provided longer answers which highlighted how differently we percieve an instrcution to think positively, we wanted to use the answers as a basis for questions used in focus group interviews, which will be the first part of my master thesis. Therefore the goal of this analysis will be to understand how people understand positive thinking, and which themes that are worth focusing on in a group discussion.

Import of necessary packages

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

Loading data

In [2]:
```
df_ma = pd.read_csv("../data/Kun_gjenta_svar.csv")
# change the file into a list
pos_answer_list = list(df_ma["Default Report"])
```

This data set shows N = 155 students answers to the open-end question "When faced with a challenge, how are you supposed to think?". Their answers were given after they recieved an encourgament to think positiviely.

In [3]:
```
# clean list, excluding instructions
pos_answer = pos_answer_list[3:]
```

relevant words

Before starting this python project, I have used Excel to find words and themes that appears several times. I have also noted variations of words (for example "positiv" and "positivt").

Process in excel:

1. Finding relevant themes by reading through every given answer
2. Condensing themes that are similiar
3. Coding every answer by theme (+1 when I think it fits the theme)

Process in python:

1. Writing for-loop code for each theme
2. Going through answers in excel to see which variations for same words that has been used
3. Adding all variations (including typos, ex. "posetivt") of words/themes to theme_list - used the total from excel step 3 to get an indications to whether I have captured most of the words relevant to the themes or not

In this next part I have made code to find how many times words appear.

In [4]:
```
# list with variation of the word "positiv":
pos_list = ["positiv","positiv ," positiv", "positivt!", "positivt ", " positivt", "positivt", "

# word_counts = {word: 0 for word in pos_list}
pos_total_count = 0

# loop through each string in the list
for string in pos_answer:
    # loop through each word in the list
    for word in pos_list:
        # count the number of times the word appears in the string
        word_count = string.count(word)
        pos_total_count += word_count

print("The total time a variation of positiv is mentioned is:", pos_total_count)
```

```
The total time a variation of positiv is mentioned is: 76
```

In [5]:
```
# list with variation of the word "not-negatively":
not_neg_list = ["Ikke se hindringer", "ikke gjøre det vanskeligere", "negativladet tanke, la meg 

# word_counts = {word: 0 for word in not_neg_list}
not_neg_total_count = 0

# loop through each string in the list
for string in pos_answer:
    # loop through each word in the list
    for word in not_neg_list:
        # count the number of times the word appears in the string
        word_count = string.count(word)
        not_neg_total_count += word_count

print("The total time a variation of not-negatively is mentioned is:", not_neg_total_count)
```

```
The total time a variation of not-negatively is mentioned is: 5
```

In [6]:
```
# list with variation of the word "negativt":
neg_list = ["motstand", "ikke ser løsningen", "vanskelig", "utfordring", "utfordrende", "problemet

# word_counts = {word: 0 for word in neg_list}
neg_total_count = 0

# loop through each string in the list
for string in pos_answer:
    # loop through each word in the list
    for word in neg_list:
        # count the number of times the word appears in the string
        word_count = string.count(word)
        neg_total_count += word_count

print("The total time a variation of negatively is mentioned is:", neg_total_count)
```

```
The total time a variation of negatively is mentioned is: 89
```

In [7]:
```
# a list with variation of the word "thinking":
think_list = ["tenk", "tenker", "logisk", "faktaene", "tenke", "strategisk", "fornuftig", "samse"

# word_counts = {word: 0 for word in think_list}
think_total_count = 0

# loop through each string in the list
for string in pos_answer:
    # loop through each word in the list
    for word in think_list:
        # count the number of times the word appears in the string
        word_count = string.count(word)
        think_total_count += word_count

print("The total time a variation of thinking is mentioned is:", think_total_count)
```

```
The total time a variation of thinking is mentioned is: 89
```

In [8]:
```
# a list with variation of the word "Solution":
sol_list = ["løsning", "løse", "løsningsorientert", "løses", "løser", "løyse", "løsninger"]

# word_counts = {word: 0 for word in think_list}
sol_total_count = 0

# loop through each string in the list
for string in pos_answer:
    # loop through each word in the list
    for word in sol_list:
        # count the number of times the word appears in the string
        word_count = string.count(word)
        sol_total_count += word_count

print("The total time a variation of solution is mentioned is:", sol_total_count)
```

```
The total time a variation of solution is mentioned is: 123
```

In [9]:
```
# a list with variation of the word "Performance":
perf_list = ["presetere", "best", "godt", "klare"]

# word_counts = {word: 0 for word in think_list}
perf_total_count = 0

# loop through each string in the list
for string in pos_answer:
    # loop through each word in the list
    for word in perf_list:
        # count the number of times the word appears in the string
        word_count = string.count(word)
        perf_total_count += word_count

print("The total time a variation of performance is mentioned is:", perf_total_count)
```

```
The total time a variation of performance is mentioned is: 44
```

In [10]:
```
# a list with variation of the word "Openness":
open_list = ["Åpent", "Åpent", "Kreativt", "kreativt", "Søregent", "interesse", "utenfor boksen",

# word_counts = {word: 0 for word in think_list}
open_total_count = 0

# loop through each string in the list
for string in pos_answer:
    # loop through each word in the list
    for word in open_list:
        # count the number of times the word appears in the string
        word_count = string.count(word)
        open_total_count += word_count

print("The total time a variation of openness is mentioned is:", open_total_count)
```

```
The total time a variation of openness is mentioned is: 47
```

In [11]:
```
# a list with variation of the word "Self-esteem":
self_list = ["jeg klare", "klarer jeg", "fikser jeg", "jeg vil finne en løsning", "jeg mestrer", "

# word_counts = {word: 0 for word in think_list}
self_total_count = 0

# loop through each string in the list
for string in pos_answer:
    # loop through each word in the list
    for word in self_list:
        # count the number of times the word appears in the string
        word_count = string.count(word)
        self_total_count += word_count

print("The total time a variation of self-esteem is mentioned is:", self_total_count)
```

```
The total time a variation of self-esteem is mentioned is: 25
```

Visualizing of results

I wanted to see which themes were the most prominent. Below is a bar chart.

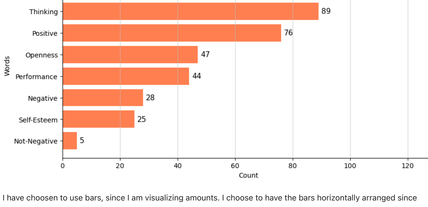
In [12]:
```
# Sort the data in ascending order based on count
data = {'Positive': pos_total_count, 'Not-Negative': not_neg_total_count, 'Negative': neg_total_c
        'Thinking': think_total_count, 'Solution': sol_total_count, 'Performance': perf_total_coun
        'Openness': open_total_count, 'Self-Esteem': self_total_count}
data_sorted = sorted(data.items(), key=lambda x: x[1])
```

In [13]:
```
# Create a horizontal bar chart for each word
fig, ax = plt.subplots(figsize=(10, 5))
plt.barh([d[0] for d in data_sorted], [d[1] for d in data_sorted], color = 'coral')
plt.title('Prevalence of Words Used to Describe Positive Thinking')
plt.xlabel('Count')
plt.ylabel('Words')

# Display the exact count next to each bar
for i, count in enumerate([d[1] for d in data_sorted]):
    plt.text(count+1, i, str(count), va='center', fontsize=11)

# Add gridlines at every 20 count interval up to 120
ax.set_xticks(range(0, 130, 20))
ax.set_yticks([d[0] for d in data_sorted])
ax.grid(axis='x', linestyle='-', alpha=0.7, color = 'silver')

# Remove frame lines
ax.spines['top'].set_visible(False)
ax.spines['bottom'].set_visible(True)
ax.spines['left'].set_visible(True)
ax.spines['right'].set_visible(False)
```



I have choosen to use bars, since I am visualizing amounts. I choose to have the bars horizontally arranged since the words were long, and this looked the neatest.

Some researchers in the field of positive thinking claim that thinking positively includes to not think negatively. Furthermore, it is claimed that positive and negative are polar concepts. However, others have criticized this, suggesting that positive thinking does not exclude negative thinking, but rather implements it. According to the results from this simple analysis the latter seems to be true.