posthink_master

This project marks the end of my forskerlinje year and the beginning of my master thesis. The data is retrived 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 differently people percieve positive thinking, and which themes that are worth focusing on in a group discussion.

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

In [2]:
```
df_ma = pd.read_csv("/Users/marieprimdahl/Documents/Studie/Programmer/GitHub/posthink_master/data/Kun_gjenta_svar.csv")
pos_answer_list = list(df_ma["Default Report"])
```

In [3]:
```
# clean list, excluding distructions
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 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", "muligheter", "engasjement", "posetivt"

# 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 endre", "snu det til noe positivt", "I

# 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: 28
```

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

# 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 'solution' is mentioned is:", perf_total_count)
```

```
The total time a variation of 'solution' is mentioned is: 44
```

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

# 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 'solution' is mentioned is:", open_total_count)
```

```
The total time a variation of 'solution' 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", "klare jeg", "blir å klare", "klarer j

# 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 'solution' is mentioned is:", self_total_count)
```

```
The total time a variation of 'solution' is mentioned is: 25
```

Visualizing of results

In [12]:
```
# Create a bar chart for each word
fig, ax = plt.subplots(figsize=(8, 5))
plt.bar(['Pos','Not-Neg','Neg', 'Think', 'Sol', 'Perf', 'Open', 'Self'], [pos_total_count, not_neg_total_count, neg_total_count, think_
plt.title('Prevalence of Words Used to Answer Positive Thinking')
plt.xlabel('Words')
plt.ylabel('Count')


# Display the exact count above each bar
for i, count in enumerate([pos_total_count, not_neg_total_count, neg_total_count, think_total_count, sol_total_count, perf_total_count,
    plt.text(i, count+1, str(count), ha='center', fontsize=12)

plt.show()
```