survey_response

May 3, 2022

0.0.1 Imports

[142]:
```
import pandas as pd
import re
import itertools
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from spacy.lang.en import English
from booknlp.booknlp import BookNLP
from IPython.core.pylabtools import figsize
sns.set_theme(style="darkgrid")
figsize(25, 20)
pd.set_option('display.max_colwidth', None)
nlp = English()
tokenizer = nlp.tokenizer
```

0.0.2 Survey data and stories

[143]:
```
def get_indexes(raw):
    if( not pd.isna(raw)):
        highlights = re.findall(r'\d+:', raw)
        return list(map(lambda x: int(x.replace(':', '')) - 1, highlights))
```

[144]:
```
data_unfiltered = pd.read_csv('survey_responses.csv')
data = data_unfiltered
# data = data_unfiltered[(data_unfiltered['aphantasia_check'] == '5') |␣
 ↪(data_unfiltered['aphantasia_check'] == '6')]
# data = data_unfiltered[(data_unfiltered['hours_read'] != '0-1 hours')]
data = data.drop(['StartDate', 'ResponseId', 'EndDate', 'IPAddress',␣
 ↪'Progress', 'Duration (in seconds)', 'RecipientLastName',␣
 ↪'RecipientFirstName', 'RecipientEmail', 'ExternalReference',␣
 ↪'LocationLatitude', 'LocationLongitude', 'DistributionChannel',␣
 ↪'Q_RecaptchaScore', 'UserLanguage'], axis=1)[data['Finished'] == 'True']
data['highlight_hp_high_1'] = data['highlight_hp_high_1'].map(get_indexes)
data['highlight_hp_high_2'] = data['highlight_hp_high_2'].map(get_indexes)
data['highlight_hp_low_1'] = data['highlight_hp_low_1'].map(get_indexes)
data['highlight_hp_low_2'] = data['highlight_hp_low_2'].map(get_indexes)
```

```
data['highlight_hero_high_1'] = data['highlight_hero_high_1'].map(get_indexes)
data['highlight_hero_high_2'] = data['highlight_hero_high_2'].map(get_indexes)
data['highlight_hero_low_1'] = data['highlight_hero_low_1'].map(get_indexes)
data['highlight_hero_low_2'] = data['highlight_hero_low_2'].map(get_indexes)
```

[145]:
```
hp_high_sentences = list(open('hp_high.txt', 'r').read().split('\n'))
hp_low_sentences = list(open('hp_low.txt', 'r').read().split('\n'))
hero_high_sentences = list(open('hero_high.txt', 'r').read().split('\n'))
hero_low_sentences = list(open('hero_low.txt', 'r').read().split('\n'))
```

[146]:
```
all_stories = ['hp_high', 'hp_low', 'hero_high', 'hero_low']
all_sentences = [hp_high_sentences, hp_low_sentences, hero_high_sentences,␣
 ↪hero_low_sentences]
```

0.0.3 Utility functions

[147]:
```
def softmax(vec):
  exponential = np.exp(vec)
  probabilities = exponential / np.sum(exponential)
  return probabilities

vector = np.array([1, 2, 3, 4, 5, 6])
probabilities = softmax(vector)
print(probabilities)
```

```
[0.00426978 0.01160646 0.03154963 0.08576079 0.23312201 0.63369132]
```

[148]:
```
def group_highlights(sent, highlights, aphantasia):
   highlight_array = np.zeros(len(sent))
   for (x, y) in zip(highlights, aphantasia):
      if(x != None):
         highlight_array[x] += 1
   return highlight_array
```

[149]:
```
def get_highlight_text(sent, bool_array):
   return [sent[i] for i in np.where(bool_array)[0]]
```

[150]:
```
def show_highlight_results(storyName, sentences, data):
   column_inc = f"highlight_{storyName}_1"
   column_dec = f"highlight_{storyName}_2"
   y1 = group_highlights(sentences, data[column_inc], data['aphantasia_check'])
   y2 = group_highlights(sentences, data[column_dec], data['aphantasia_check'])

   return (y1, y2)
```

[151]:
```
def get_sentence(num, story):
   story_tokens = tokens[story]
```

```
    row = story_tokens[story_tokens['token_ID_within_document'] ==␣
↪num]['sentence_ID']
    if(len(row) > 0):
        return story_tokens[story_tokens['token_ID_within_document'] ==␣
↪num]['sentence_ID'].iloc[0]
    else:
        return -1
```

[152]:
```
def get_supersense(story):
    supersense_df = pd.read_csv(f"results/{story}/{story}.supersense",␣
↪delimiter='\t')
    story_list = list(itertools.repeat(story, len(supersense_df['start_token'])))
    supersense_df['sentence'] = list(map(get_sentence,␣
↪supersense_df['start_token'], story_list))
    return supersense_df
```

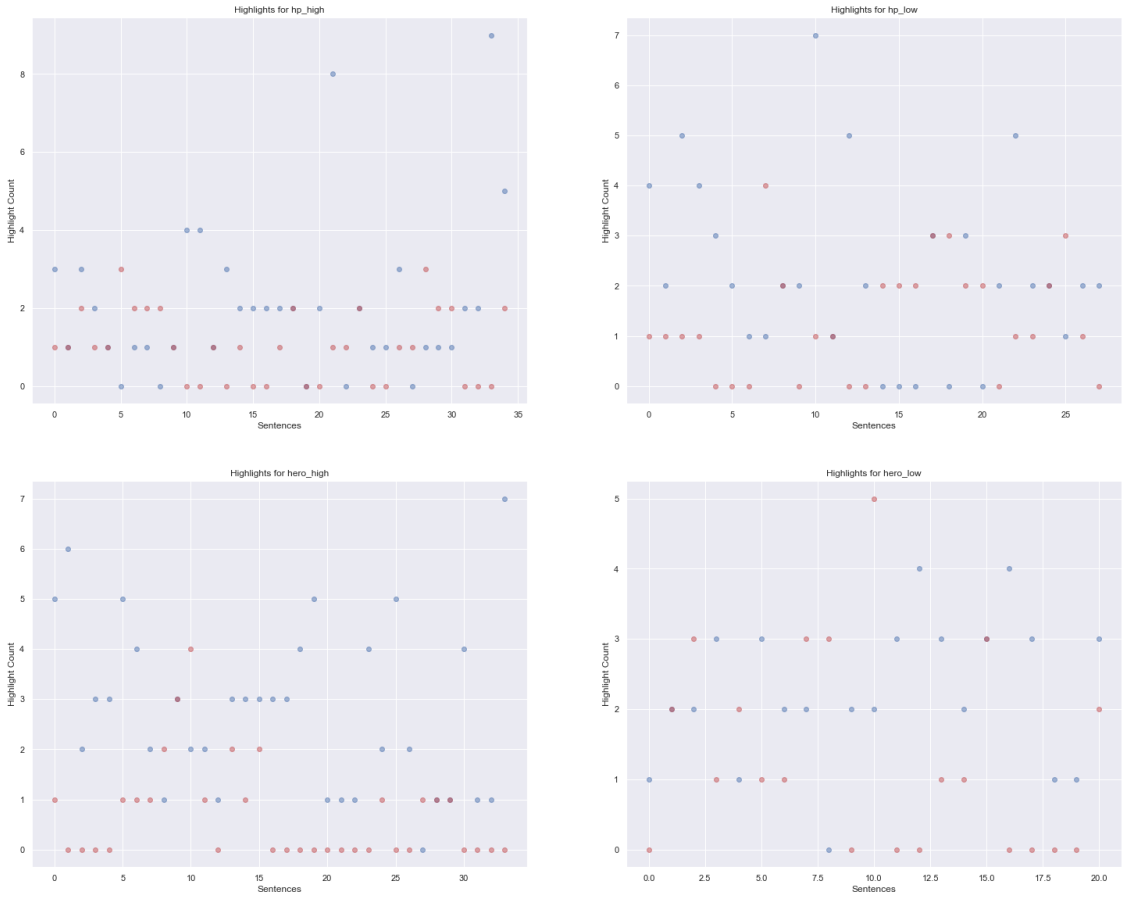
[153]:
```
def count_cognition_words(supersense, sent):
    cognition_counts = np.zeros(len(sent))
    for i in range(len(sent)):
        supersense_tokens = supersense[supersense['sentence'] ==␣
↪i]['supersense_category']
        for j in supersense_tokens:
            if(j == 'noun.cognition' or j == 'verb.cognition'):
                cognition_counts[i] += 1
        cognition_counts[i] = (cognition_counts[i] /␣
↪len(list(tokenizer(sent[i]))))*100
    return cognition_counts
```

[154]:
```
def get_concreteness_scores(sentence):
    tokens = tokenizer(sentence)
    score = 0
    for token in tokens:
        row = concreteness_scores[concreteness_scores['Word'] == str(token)]
        if(len(row) > 0):
            score = score + row['Conc.M'].iloc[0]
    return score / len(tokens)
```

[155]:
```
def get_overlapping_highlights(inc, dec):
    overlaps = []
    for row in range(len(inc)):
        if inc[row] != 0 and dec[row] != 0:
            overlap = inc[row] if inc[row] < abs(dec[row]) else abs(dec[row])
            overlaps.append(overlap)
        else:
            overlaps.append(0)
    return overlaps
```

0.0.4 Process highlights

[156]:
```
counts_inc = dict(zip(all_stories, np.zeros(len(all_stories))))
counts_dec = dict(zip(all_stories, np.zeros(len(all_stories))))
sentence_counts = dict(zip(all_stories, np.zeros(len(all_stories))))
combined_highlights = dict(zip(all_stories, np.zeros(len(all_stories))))
overlapping_highlights = dict(zip(all_stories, np.zeros(len(all_stories))))
sentences = dict(zip(all_stories, all_sentences))

fig, ax = plt.subplots(2, 2)
count = 0
axes = [[0, 0], [0, 1], [1, 0], [1, 1]]

for story in all_stories:
    counts_inc[story], counts_dec[story] = show_highlight_results(story,␣
 ↪sentences[story], data) # show individual results by passing in a row:␣
 ↪data[1:2]
    sentence_counts[story] = pd.DataFrame(sentences[story], columns=['sentence'])
    sentence_counts[story]['increase'] = counts_inc[story]
    sentence_counts[story]['decrease'] = counts_dec[story]
    sentence_counts[story]['combined'] = counts_inc[story] - counts_dec[story]
    combined_highlights[story] = pd.DataFrame(sentences[story],␣
 ↪columns=['sentence'])
    combined_highlights[story]['combined_highlights'] = counts_inc[story] -␣
 ↪counts_dec[story]
    overlapping_highlights[story] = pd.DataFrame(sentences[story])
    overlapping_highlights[story]['overlap'] =␣
 ↪get_overlapping_highlights(counts_inc[story], counts_dec[story])

    # display(sentence_counts[story].head())
    # Plot the highlights
    x = np.linspace(0, len(sentences[story]) -1, len(sentences[story]))
    ax1 = axes[count][0]
    ax2 = axes[count][1]
    ax[ax1, ax2].scatter(x, counts_inc[story], alpha=.5, color='b')
    ax[ax1, ax2].scatter(x, counts_dec[story], alpha=.5, color='r')
    ax[ax1, ax2].set_xlabel("Sentences")
    ax[ax1, ax2].set_ylabel("Highlight Count")
    ax[ax1, ax2].set_title(f"Highlights for {all_stories[count]}")

    count +=1
plt.show()
```



[157]:
```
for story in all_stories:
    display(overlapping_highlights[story].sort_values(by='overlap',␣
 ↪ascending=False).head(10))
```

```
                                                                    ␣
 ↪                                                      0  \
34                                                                  ␣
 ↪       Hagrid stood in the doorway, looking very impressive.
2                                                                   ␣
 ↪                      The one that Sirius had fallen behind.
23                                                                  ␣
 ↪                        Dudley, Where had the Veil sent him?
18                                                                  ␣
 ↪       Why, even at that very moment, he couldn't walk away.
12                                              Cedric Sirius Dumbledore Hedwig␣
 ↪Moody Dobby Tonks father Remus Colin Creevy Tonks Snape Fred.
30  Although it did occur to Harry that when people were trying to break in, it␣
 ↪can generally be assumed that they are probably armed as well.
```

```
29                                                  His uncle didn't seem to know␣
 ↪that, though, and shouted "Who's there, I warn you-I'm armed!"
28                                                                               ␣
 ↪                  Like that would do any good against HAGRID.
26                                                  There was a crash behind them␣
 ↪and Uncle Vernon came skidding into the room, holding a rifle.
21              He was lying on a floor, covered by a poor excuse for a blanket,␣
 ↪and someone appeared to be attempting to break the door down.

    overlap
34      2.0
2       2.0
23      2.0
18      2.0
12      1.0
30      1.0
29      1.0
28      1.0
26      1.0
21      1.0

                                                                                 ␣
 ↪                                                  0  \
17          Most of my concentration that evening was on keeping Bella from␣
 ↪marching over to the Gryffindor table and making a scene.
8                                I was not about to let a Bellatrix Black temper␣
 ↪tantrum ruin the glow of my first night as a prefect.
19                That "discussion" was the reason for my presence outside of␣
 ↪Gryffindor Tower on the first morning of the new term;.
24                                    I warned Sirius of Bella's approach, and␣
 ↪Sirius managed to avoid her until late in the afternoon.
1                                                                                ␣
 ↪               I looked down the Slytherin table;.
22                                                                  He was flanked␣
 ↪by three boys, none of whom I paid much attention to.
11                                                                               ␣
 ↪         Professor McGonagall was no fan of Bella's.
10  Bella continued to mutter angrily and slightly threateningly under her␣
 ↪breath until Professor McGonagall shot her a filthy look;.
0                                                             "What the hell just␣
 ↪happened?" Bella hissed at me from across the table.
7                                                                             I␣
 ↪know just how irritating I was being, and I stand by it.

    overlap
17      3.0
8       2.0
```

19 2.0
24 2.0
1 1.0
22 1.0
11 1.0
10 1.0
0 1.0
7 1.0

0 \

9 He still had a few posters up from some of his other favorite heroes, but he'd changed his sheets to a plain black set that he'd found collecting dust in the linen closet paired with a gold pillowcase.

15 He'd found it several years earlier and quickly figured out that a lot of underground heroes used it to communicate with each other, since it offered encrypted chats and accounts were only known by random numbers, rather than usernames.

13 He had thought it would be painful and it was in some ways, but it was mostly familiar, like his analysis was the only part of himself that hadn't shattered alongside his hero dream.

10 It wasn't official merchandise, but it reminded Izuku of Eraserhead anyway.

0 Even after an hour-long shower, Izuku still felt the weight of the past 24 hours clinging to his skin.

24 Izuku was hyperventilating.

28 Not quite, because the bullying had gotten worse since the sludge villain incident, but it came close!

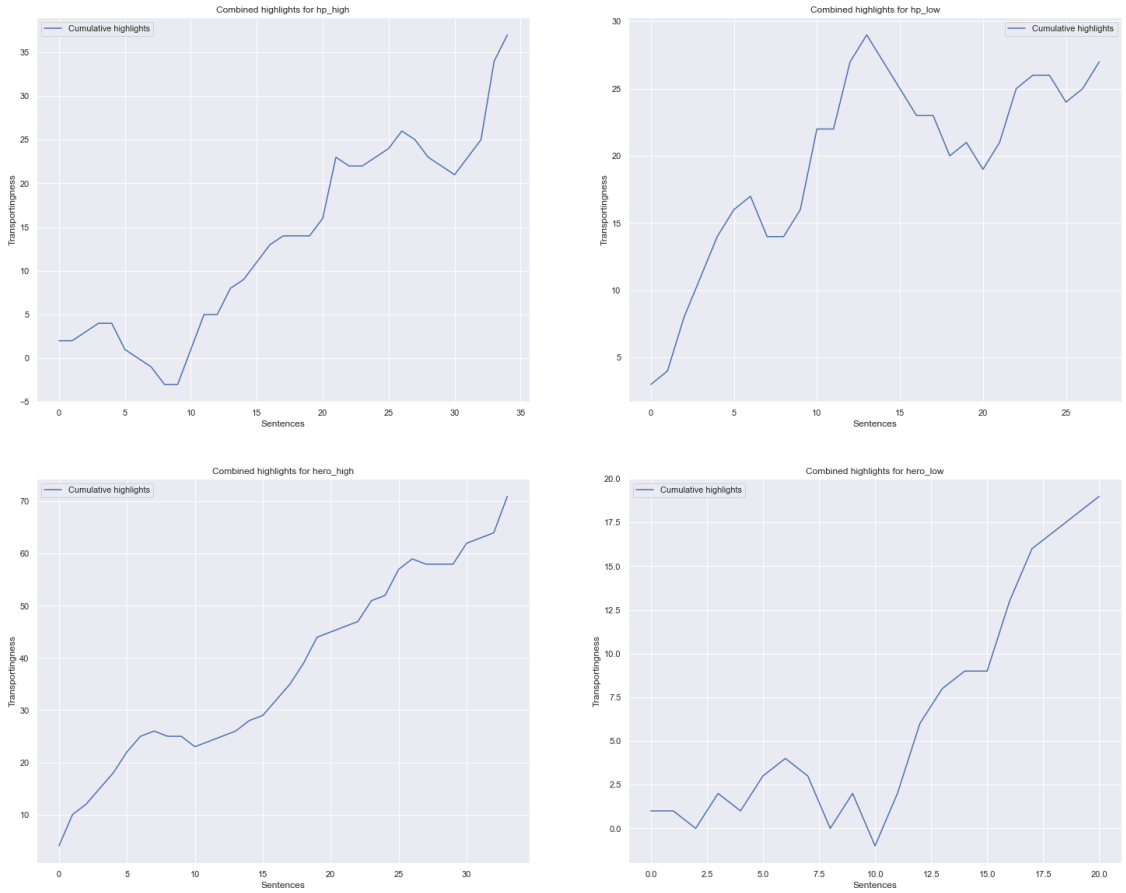
14 That was why he found himself, during his latest bout with insomnia, browsing a lesser known hero forum.

```
29                                                                  Several␣
 ↪other heroes had also sent him messages thanking him, while others, who had␣
 ↪apparently heard of his analyses from their friends, had asked him to send␣
 ↪them theirs.
11                                                          It had taken a few␣
 ↪days for him to gather the courage, but Izuku had finally sat down and updated␣
 ↪his hero analysis notebooks, leaving out the parts on All Might's secret␣
 ↪weakness.

    overlap
9       3.0
15      2.0
13      2.0
10      2.0
0       1.0
24      1.0
28      1.0
14      1.0
29      1.0
11      1.0

                                                                            ␣
 ↪                                                                          ␣
 ↪                                                                          ␣
 ↪                                                                          ␣
 ↪  0  \
15                                                                          ␣
 ↪                                                 The man held out the remote and␣
 ↪seemed to shout something to the hero from the roof, but the hero continued to␣
 ↪scale the wall and shot himself all the way to the top with a large blast of␣
 ↪fire.
10  Kumiko bought a simple cap to cover her hair at work (her disguises only␣
 ↪lasted for so long without training and she had very little endurance, and she␣
 ↪soon found out that they began dissolving from top down - thus the hat), and␣
 ↪Himiko bought herself a cheap winter coat, as it was starting to get chilly in␣
 ↪the city.
7                                                                           ␣
 ↪                                                                          ␣
 ↪                      She didn't know his quirk, but she just told the␣
 ↪shopkeeper that it was a minor quirk that let him see slightly better and got␣
 ↪away with it.
1                                                                           ␣
 ↪                                                                          ␣
 ↪   First, her older sister was kicked out of their own house unjustly by her␣
 ↪parents, so she ran away from home to accompany her and make sure Himiko␣
 ↪didn't die.
```

```
20                                                                              ␣
 ↪                                                                              ␣
 ↪                                  Himiko stumbled, her hand slipping away from␣
 ↪Kumiko's, and she screamed for Himiko as she dove out of the way of the␣
 ↪falling debris.
2                                                                               ␣
 ↪                                                                              ␣
 ↪                                      She knew that if she returned, her parents␣
 ↪would punish her for running away, and that was not exactly the most ideal␣
 ↪situation.
5                                                                               ␣
 ↪                               They set up a shelter tucked into an alleyway using␣
 ↪anything waterproof or durable that they could find in a dumpster nearby, and␣
 ↪luckily they had also found some old blankets and fabric they could use for␣
 ↪bedding.
6                                                                               ␣
 ↪                                                                              ␣
 ↪                          A few months from then, Kumiko managed to get a job␣
 ↪moving boxes for a nearby shop, disguised as a young boy she'd seen on the␣
 ↪streets once.
4                                                                               ␣
 ↪       Himiko proposed that they steal food and other necessities from stores,␣
 ↪but Kumiko didn't like the idea of stealing from smaller stores and managed to␣
 ↪convince her sister to only steal from people who could afford to lose some␣
 ↪money.
3                                                                               ␣
 ↪                                                                              ␣
 ↪                                                                              ␣
 ↪                  This caused some problems: no money, no shelter, no food or␣
 ↪water.

    overlap
15      3.0
10      2.0
7       2.0
1       2.0
20      2.0
2       2.0
5       1.0
6       1.0
4       1.0
3       1.0
```

[158]:
```
fig, ax = plt.subplots(2, 2)
count = 0

for story in all_stories:
```

```
    cdf = np.cumsum(sentence_counts[story]['combined'])
    # Plot the highlights
    x = np.linspace(0, len(sentences[story]) -1, len(sentences[story]))
    ax1 = axes[count][0]
    ax2 = axes[count][1]
    ax[ax1, ax2].plot(x, cdf, color='b', label="Cumulative highlights")
    ax[ax1, ax2].set_xlabel("Sentences")
    ax[ax1, ax2].set_ylabel("Transportingness")
    ax[ax1, ax2].set_title(f"Combined highlights for {all_stories[count]}")
    ax[ax1, ax2].legend()
    count +=1

plt.show()
```



[159]:
```
for story in all_stories:
    df = combined_highlights[story]
    display(df.style.background_gradient(axis=0, gmap=df['combined_highlights'],␣
 ↪cmap='RdBu',  vmin=-2.0, vmax=4.0))
```

```
<pandas.io.formats.style.Styler at 0x29ecfa460>
```

```
<pandas.io.formats.style.Styler at 0x29ecfa460>
```

```
<pandas.io.formats.style.Styler at 0x29ecfa460>
```

```
<pandas.io.formats.style.Styler at 0x29ecfa460>
```

[160]:
```
questions = pd.read_csv('survey_responses.csv')[0:1]
```

[161]:
```
scale_questions = ['emotional_affect', 'forgetting_surroundings', 'distracted',␣
 ↪'characters_alive', 'mental_imagery']
```

[162]:
```
def compute_transportation(story):
    ratings = []
    scale = ['Strongly disagree', 'Slightly disagree', 'Neutral', 'Slightly␣
 ↪agree', 'Strongly agree']
    ratings.append(np.round(np.mean(list(map(lambda x: scale.index(x),␣
 ↪data[f'rate_{story}_1']))), 2))
    ratings.append(np.round(np.mean(list(map(lambda x: scale.index(x),␣
 ↪data[f'rate_{story}_2']))), 2))
    ratings.append(np.round(np.mean(list(map(lambda x: scale.index(x),␣
 ↪data[f'rate_{story}_3']))), 2))
    ratings.append(np.round(np.mean(list(map(lambda x: scale.index(x),␣
 ↪data[f'rate_{story}_4']))), 2))
    ratings.append(np.round(np.mean(list(map(lambda x: scale.index(x),␣
 ↪data[f'rate_{story}_5']))), 2))
    score = ratings[0] + ratings[1] - ratings[2] + ratings[3] + ratings[4]
    return (score, ratings)
```

[163]:
```
# highest possible score: 16
transportation_scores = dict(zip(all_stories, np.zeros(len(all_stories))))
transportation_ratings = dict(zip(all_stories, np.zeros(len(all_stories))))
transportation_ratings_human = dict(zip(all_stories, np.
 ↪zeros(len(all_stories))))

for story in all_stories:
    transportation_scores[story], transportation_ratings[story] =␣
 ↪compute_transportation(story)
    transportation_ratings_human[story] = pd.DataFrame(dict(zip(scale_questions,␣
 ↪transportation_ratings[story])).items(), columns=['Question', 'Rating'])
    print(f'Transportation score for {story}: {np.
 ↪round(transportation_scores[story], 2)}, average ratings:')
    display(transportation_ratings_human[story])
```

```
Transportation score for hp_high: 8.3, average ratings:
```

	Question	Rating
0	emotional_affect	2.1
1	forgetting_surroundings	2.5
2	distracted	2.1

```
3         characters_alive     3.3
4           mental_imagery     2.5
```

```
Transportation score for hp_low: 7.0, average ratings:
```

```
                  Question  Rating
0         emotional_affect     1.6
1  forgetting_surroundings     2.5
2               distracted     2.2
3         characters_alive     2.4
4           mental_imagery     2.7
```

```
Transportation score for hero_high: 8.1, average ratings:
```

```
                  Question  Rating
0         emotional_affect     2.3
1  forgetting_surroundings     2.6
2               distracted     1.6
3         characters_alive     2.4
4           mental_imagery     2.4
```

```
Transportation score for hero_low: 5.6, average ratings:
```

```
                  Question  Rating
0         emotional_affect     1.9
1  forgetting_surroundings     2.2
2               distracted     2.5
3         characters_alive     1.8
4           mental_imagery     2.2
```

0.0.5 BookNLP, LIWC, and Concreteness data

[164]:
```
def run_booknlp(story):
        model_params={
                        "pipeline":"entity,quote,supersense,event,coref",
                        "model":"big"
                }

        booknlp=BookNLP("en", model_params)

        # Input file to process
        input_file=f"{story}.txt"

        # Output directory to store resulting files in
        output_directory=f"results/{story}"

        # File within this directory will be named ${book_id}.entities,␣
 ↪${book_id}.tokens, etc.
        book_id=story
```

```
        booknlp.process(input_file, output_directory, book_id)
```

[165]:
```
# for story in all_stories:
#     run_booknlp(story)
```

[166]:
```
concreteness_scores = pd.read_csv("concreteness_scores.csv")
liwc = pd.read_csv('liwc_results.csv')
```

[167]:
```
import textstat
def readability(sent):
    return textstat.text_standard(sent, float_output=True)
    # return (1 if grade_level > 8.0 else 0)
```

[168]:
```
tokens = dict(zip(all_stories, np.zeros(len(all_stories))))
supersense = dict(zip(all_stories, np.zeros(len(all_stories))))
sentiment = dict(zip(all_stories, np.zeros(len(all_stories))))
pairplot_df = dict(zip(all_stories, np.zeros(len(all_stories))))

for story in all_stories:
    tokens[story] = pd.read_csv(f"results/{story}/{story}.tokens",␣
 ↪delimiter='\t')
    supersense[story] = get_supersense(story)
    sentiment[story] = pd.read_csv(f"feature_results/sentiment/{story}_sent.
 ↪csv")['sentiment']
    sentence_counts[story]['cognition'] =␣
 ↪count_cognition_words(supersense[story], sentences[story])
    sentence_counts[story]['perception'] = list(liwc[liwc['Filename'] ==␣
 ↪f'{story}.txt']['Perception'])
    sentence_counts[story]['concreteness'] = list(map(get_concreteness_scores,␣
 ↪sentences[story]))
    sentence_counts[story]['sentiment'] = list(sentiment[story])
    sentence_counts[story]['sentiment_binary'] = list(map(lambda x: 1 if abs(x)␣
 ↪> 0 else 0, (sentiment[story])))
    sentence_counts[story]['complexity'] = list(map(readability,␣
 ↪sentences[story]))
    pairplot_df[story] = sentence_counts[story].drop(['sentence'], axis=1)
```

0.0.6 Compute Correlations

[169]:
```
# story_pairplot = sns.pairplot(pairplot_df['hero_high'])
# story_pairplot.fig.suptitle(f"Pairplot for hero_high")
# plt.show()
```

[170]:
```
# Normalize data for correlation checking
def absolute_maximum_scale(series):
    return series / series.abs().max()
```

```
pairplot_norm = dict(zip(all_stories, np.zeros(len(all_stories))))

for story in all_stories:
    pairplot_norm[story] = pd.DataFrame(pairplot_df[story])
    for col in pairplot_norm[story].drop(['increase', 'decrease', 'combined'],␣
 ↪axis=1).columns:
        pairplot_norm[story][col] =␣
 ↪absolute_maximum_scale(pairplot_norm[story][col])
```

[171]:
```
corr = dict(zip(all_stories, np.zeros(len(all_stories))))
f, ax = plt.subplots(2, 2)
count = 0
axes = [[0, 0], [0, 1], [1, 0], [1, 1]]
f.tight_layout()

for story in all_stories:
    ax1 = axes[count][0]
    ax2 = axes[count][1]

    # Compute the correlation matrix
    corr[story] = pairplot_norm[story].drop(['combined'], axis=1).corr()

    # Generate a mask for the upper triangle
    mask = np.zeros_like(corr[story], dtype=bool)
    mask[np.triu_indices_from(mask)] = True

    # Generate a custom diverging colormap
    cmap = sns.diverging_palette(220, 10, as_cmap=True)
    # Draw the heatmap with the mask and correct aspect ratio
    sns.heatmap(
        corr[story],
        mask=mask,
        cmap=cmap,
        linewidths=0.5,
        cbar_kws={"shrink": 0.5},
        ax=ax[ax1, ax2]
    )
    ax[ax1, ax2].set_title(f"Correlation heatmap for {story}")
    print(f"Correlation for {story}")
    display(corr[story])
    count += 1

plt.show()
```

```
Correlation for hp_high
```

	increase	decrease	cognition	perception	concreteness	\
increase	1.000000	-0.255566	-0.063586	0.166053	0.544731	

```
decrease         -0.255566  1.000000  -0.056565    0.102251     -0.015454
cognition        -0.063586 -0.056565   1.000000   -0.302749      0.040994
perception        0.166053  0.102251  -0.302749    1.000000      0.172344
concreteness      0.544731 -0.015454   0.040994    0.172344      1.000000
sentiment        -0.187882  0.113765   0.140982    0.022707     -0.078300
sentiment_binary  0.415400  0.170839   0.132945   -0.116453      0.183212
complexity        0.465111  0.036398   0.331161   -0.022052      0.508606

                  sentiment  sentiment_binary  complexity
increase          -0.187882          0.415400    0.465111
decrease           0.113765          0.170839    0.036398
cognition          0.140982          0.132945    0.331161
perception         0.022707         -0.116453   -0.022052
concreteness      -0.078300          0.183212    0.508606
sentiment          1.000000         -0.238119   -0.064747
sentiment_binary  -0.238119          1.000000    0.580913
complexity        -0.064747          0.580913    1.000000
```

Correlation for hp_low

```
                  increase  decrease  cognition  perception  concreteness  \
increase          1.000000 -0.354405  -0.375455    0.173430      0.105606
decrease         -0.354405  1.000000   0.054461   -0.398019      0.173963
cognition        -0.375455  0.054461   1.000000   -0.316544     -0.185616
perception        0.173430 -0.398019  -0.316544    1.000000      0.427456
concreteness      0.105606  0.173963  -0.185616    0.427456      1.000000
sentiment        -0.322338 -0.142256   0.424105   -0.220874     -0.319147
sentiment_binary  0.071821  0.214429   0.163013   -0.096873      0.022580
complexity        0.265414  0.143607  -0.220694   -0.315167      0.005143

                  sentiment  sentiment_binary  complexity
increase          -0.322338          0.071821    0.265414
decrease          -0.142256          0.214429    0.143607
cognition          0.424105          0.163013   -0.220694
perception        -0.220874         -0.096873   -0.315167
concreteness      -0.319147          0.022580    0.005143
sentiment          1.000000         -0.212148   -0.010990
sentiment_binary  -0.212148          1.000000    0.089243
complexity        -0.010990          0.089243    1.000000
```

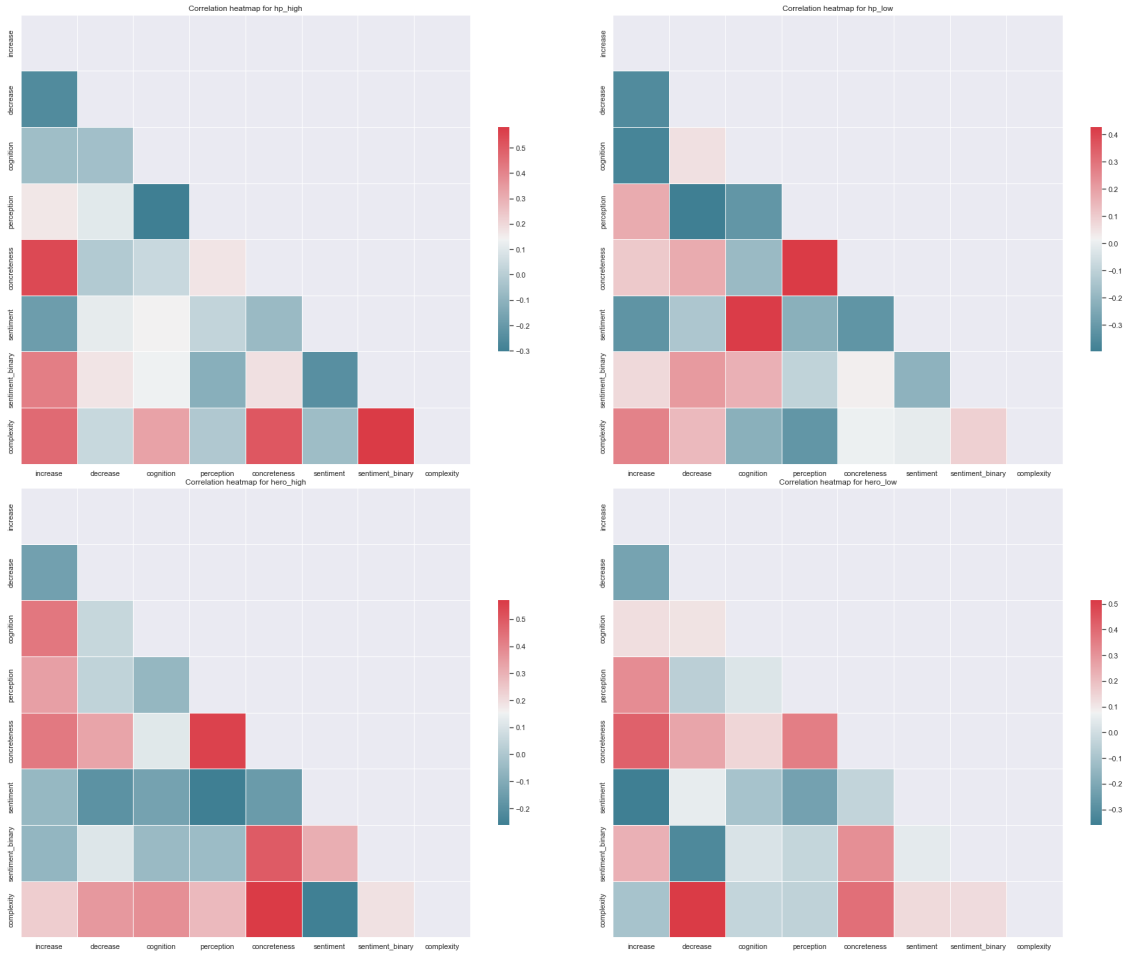
Correlation for hero_high

```
                  increase  decrease  cognition  perception  concreteness  \
increase          1.000000 -0.139494   0.433371    0.339837      0.425900
decrease         -0.139494  1.000000   0.056510    0.042629      0.332065
cognition         0.433371  0.056510   1.000000   -0.057617      0.116370
perception        0.339837  0.042629  -0.057617    1.000000      0.551471
concreteness      0.425900  0.332065   0.116370    0.551471      1.000000
sentiment        -0.053869 -0.191138  -0.133984   -0.261535     -0.160523
sentiment_binary -0.062500  0.111739  -0.047287   -0.043862      0.495056
```

```
complexity        0.236228  0.355761   0.375083    0.281505      0.569781

                  sentiment  sentiment_binary  complexity
increase          -0.053869         -0.062500    0.236228
decrease          -0.191138          0.111739    0.355761
cognition         -0.133984         -0.047287    0.375083
perception        -0.261535         -0.043862    0.281505
concreteness      -0.160523          0.495056    0.569781
sentiment          1.000000          0.307893   -0.257048
sentiment_binary   0.307893          1.000000    0.190069
complexity        -0.257048          0.190069    1.000000
```

Correlation for hero_low

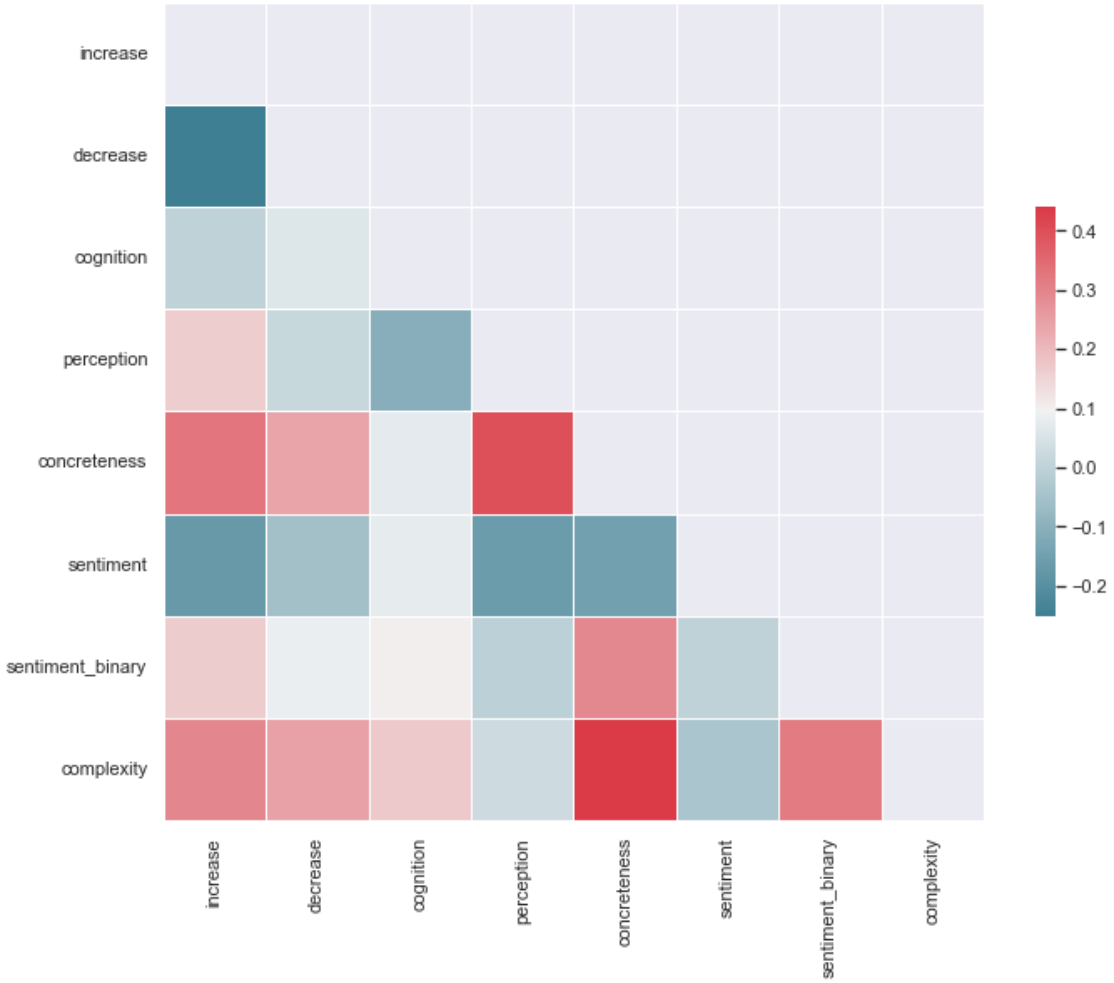
```
                  increase  decrease  cognition  perception  concreteness  \
increase          1.000000 -0.223855   0.122152    0.317234      0.420122
decrease         -0.223855  1.000000   0.111451   -0.054785      0.258413
cognition         0.122152  0.111451   1.000000    0.027372      0.143312
perception        0.317234 -0.054785   0.027372    1.000000      0.350709
concreteness      0.420122  0.258413   0.143312    0.350709      1.000000
sentiment        -0.360392  0.054086  -0.101995   -0.228583     -0.036960
sentiment_binary  0.232325 -0.319527   0.020946   -0.032327      0.310537
complexity       -0.101805  0.513034  -0.033909   -0.043938      0.383512

                  sentiment  sentiment_binary  complexity
increase          -0.360392          0.232325   -0.101805
decrease           0.054086         -0.319527    0.513034
cognition         -0.101995          0.020946   -0.033909
perception        -0.228583         -0.032327   -0.043938
concreteness      -0.036960          0.310537    0.383512
sentiment          1.000000          0.046117    0.132239
sentiment_binary   0.046117          1.000000    0.129865
complexity         0.132239          0.129865    1.000000
```



[172]:
```
drop_cols = ['combined']
stories_concat = pd.concat( [pairplot_norm['hero_high'].drop(drop_cols, axis=1),
pairplot_norm['hero_low'].drop(drop_cols, axis=1),
pairplot_norm['hp_high'].drop(drop_cols, axis=1),
pairplot_norm['hp_low'].drop(drop_cols, axis=1)], axis=0)
# stories_concat = pairplot_df['hp_high']
# Compute the correlation matrix
corr = stories_concat.corr()

# Generate a mask for the upper triangle
mask = np.zeros_like(corr, dtype=bool)
mask[np.triu_indices_from(mask)] = True

# Generate a custom diverging colormap
cmap = sns.diverging_palette(220, 10, as_cmap=True)

# Set up the matplotlib figure
```

```
f, ax = plt.subplots(figsize=(11, 9))

sns.heatmap(
    corr,
    mask=mask,
    cmap=cmap,
    linewidths=0.5,
    cbar_kws={"shrink": 0.5},
    ax=ax
)
display(corr)

plt.show()
```

```
                 increase  decrease  cognition  perception  concreteness  \
increase         1.000000 -0.250489  -0.000544    0.163337      0.327935
decrease        -0.250489  1.000000   0.058907    0.014119      0.240554
cognition       -0.000544  0.058907   1.000000   -0.109354      0.070500
perception       0.163337  0.014119  -0.109354    1.000000      0.398840
concreteness     0.327935  0.240554   0.070500    0.398840      1.000000
sentiment       -0.171458 -0.054175   0.075261   -0.162563     -0.150260
sentiment_binary 0.166253  0.083486   0.101000   -0.006855      0.293020
complexity       0.296809  0.244079   0.172263    0.027087      0.439782

                 sentiment  sentiment_binary  complexity
increase         -0.171458          0.166253    0.296809
decrease         -0.054175          0.083486    0.244079
cognition         0.075261          0.101000    0.172263
perception       -0.162563         -0.006855    0.027087
concreteness     -0.150260          0.293020    0.439782
sentiment         1.000000         -0.000904   -0.041835
sentiment_binary -0.000904          1.000000    0.318408
complexity       -0.041835          0.318408    1.000000
```


increase
decrease
cognition
perception
concreteness
sentiment
sentiment_binary
complexity
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2

[ ]: