2_1_Lexical_Complexity_Binary_Classification_Prediction—Data_Preparation_V2

April 11, 2025

[1]:
```
#@title Install Packages
```

[2]:
```
!pip install -q transformers
!pip install -q torchinfo
!pip install -q datasets
!pip install -q evaluate
!pip install -q nltk
!pip install -q contractions
```

```
                                        491.2/491.2 kB
26.8 MB/s eta 0:00:00
                                        116.3/116.3 kB
12.3 MB/s eta 0:00:00
                                        183.9/183.9 kB
6.2 MB/s eta 0:00:00
                                        143.5/143.5 kB
11.0 MB/s eta 0:00:00
                                        194.8/194.8 kB
15.7 MB/s eta 0:00:00
```

```
ERROR: pip's dependency resolver does not currently take into account
all the packages that are installed. This behaviour is the source of the
following dependency conflicts.
torch 2.6.0+cu124 requires nvidia-cublas-cu12==12.4.5.8; platform_system ==
"Linux" and platform_machine == "x86_64", but you have nvidia-cublas-cu12
12.5.3.2 which is incompatible.
torch 2.6.0+cu124 requires nvidia-cuda-cupti-cu12==12.4.127; platform_system ==
"Linux" and platform_machine == "x86_64", but you have nvidia-cuda-cupti-cu12
12.5.82 which is incompatible.
torch 2.6.0+cu124 requires nvidia-cuda-nvrtc-cu12==12.4.127; platform_system ==
"Linux" and platform_machine == "x86_64", but you have nvidia-cuda-nvrtc-cu12
12.5.82 which is incompatible.
torch 2.6.0+cu124 requires nvidia-cuda-runtime-cu12==12.4.127; platform_system
== "Linux" and platform_machine == "x86_64", but you have nvidia-cuda-runtime-
cu12 12.5.82 which is incompatible.
torch 2.6.0+cu124 requires nvidia-cudnn-cu12==9.1.0.70; platform_system ==
"Linux" and platform_machine == "x86_64", but you have nvidia-cudnn-cu12
9.3.0.75 which is incompatible.
torch 2.6.0+cu124 requires nvidia-cufft-cu12==11.2.1.3; platform_system ==
"Linux" and platform_machine == "x86_64", but you have nvidia-cufft-cu12
11.2.3.61 which is incompatible.
torch 2.6.0+cu124 requires nvidia-curand-cu12==10.3.5.147; platform_system ==
"Linux" and platform_machine == "x86_64", but you have nvidia-curand-cu12
10.3.6.82 which is incompatible.
torch 2.6.0+cu124 requires nvidia-cusolver-cu12==11.6.1.9; platform_system ==
"Linux" and platform_machine == "x86_64", but you have nvidia-cusolver-cu12
11.6.3.83 which is incompatible.
torch 2.6.0+cu124 requires nvidia-cusparse-cu12==12.3.1.170; platform_system ==
"Linux" and platform_machine == "x86_64", but you have nvidia-cusparse-cu12
12.5.1.3 which is incompatible.
torch 2.6.0+cu124 requires nvidia-nvjitlink-cu12==12.4.127; platform_system ==
"Linux" and platform_machine == "x86_64", but you have nvidia-nvjitlink-cu12
12.5.82 which is incompatible.
gcsfs 2025.3.2 requires fsspec==2025.3.2, but you have fsspec 2024.12.0 which is
incompatible.
```


84.0/84.0 kB

```
5.8 MB/s eta 0:00:00
                                    289.9/289.9 kB
16.9 MB/s eta 0:00:00
                                    118.3/118.3 kB
9.0 MB/s eta 0:00:00
```

[3]:
```
!sudo apt-get update
! sudo apt-get install tree
```

```
Get:1 https://cloud.r-project.org/bin/linux/ubuntu jammy-cran40/ InRelease
[3,632 B]
Hit:2 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_64
InRelease
Get:3 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Hit:4 http://archive.ubuntu.com/ubuntu jammy InRelease
Get:5 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:6 https://r2u.stat.illinois.edu/ubuntu jammy InRelease [6,555 B]
Get:7 https://r2u.stat.illinois.edu/ubuntu jammy/main amd64 Packages [2,690 kB]
Hit:8 http://archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:9 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages
[2,788 kB]
Get:10 https://ppa.launchpadcontent.net/deadsnakes/ppa/ubuntu jammy InRelease
[18.1 kB]
Get:11 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [3,099
kB]
Hit:12 https://ppa.launchpadcontent.net/graphics-drivers/ppa/ubuntu jammy
InRelease
Hit:13 https://ppa.launchpadcontent.net/ubuntugis/ppa/ubuntu jammy InRelease
Get:14 https://ppa.launchpadcontent.net/deadsnakes/ppa/ubuntu jammy/main amd64
Packages [34.3 kB]
Get:15 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages
[1,542 kB]
Get:16 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages
[1,243 kB]
Get:17 https://r2u.stat.illinois.edu/ubuntu jammy/main all Packages [8,833 kB]
Fetched 20.5 MB in 4s (4,710 kB/s)
Reading package lists… Done
W: Skipping acquire of configured file 'main/source/Sources' as repository
'https://r2u.stat.illinois.edu/ubuntu jammy InRelease' does not seem to provide
it (sources.list entry misspelt?)
Reading package lists… Done
Building dependency tree… Done
Reading state information… Done
The following NEW packages will be installed:
  tree
0 upgraded, 1 newly installed, 0 to remove and 32 not upgraded.
Need to get 47.9 kB of archives.
```

```
After this operation, 116 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy/universe amd64 tree amd64 2.0.2-1
[47.9 kB]
Fetched 47.9 kB in 1s (72.0 kB/s)
debconf: unable to initialize frontend: Dialog
debconf: (No usable dialog-like program is installed, so the dialog based
frontend cannot be used. at /usr/share/perl5/Debconf/FrontEnd/Dialog.pm line 78,
<> line 1.)
debconf: falling back to frontend: Readline
debconf: unable to initialize frontend: Readline
debconf: (This frontend requires a controlling tty.)
debconf: falling back to frontend: Teletype
dpkg-preconfigure: unable to re-open stdin:
Selecting previously unselected package tree.
(Reading database … 126315 files and directories currently installed.)
Preparing to unpack …/tree_2.0.2-1_amd64.deb …
Unpacking tree (2.0.2-1) …
Setting up tree (2.0.2-1) …
Processing triggers for man-db (2.10.2-1) …
```

[4]:
```
#@title Imports
import nltk
from nltk.tokenize import RegexpTokenizer

import evaluate
import transformers

import contractions

from torchinfo import summary
from datasets import load_dataset

from transformers import AutoTokenizer, AutoModel,␣
 ↪AutoModelForSequenceClassification
from transformers import TrainingArguments, Trainer

import os
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

import sklearn

import spacy
```

[5]:
```
# @title Mount Google Drive
```

[6]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

[7]:
```
dir_root = '/content/drive/MyDrive/266-final/'
# dir_data = '/content/drive/MyDrive/266-final/data/'
# dir_data = '/content/drive/MyDrive/266-final/data/se21-t1-comp-lex-master/'
dir_data = '/content/drive/MyDrive/266-final/data/266-comp-lex-master'
dir_models = '/content/drive/MyDrive/266-final/models/'
dir_results = '/content/drive/MyDrive/266-final/results/'
```

[8]:
```
!tree /content/drive/MyDrive/266-final/data/266-comp-lex-master/
```

```
/content/drive/MyDrive/266-final/data/266-comp-lex-master/
  fe-test-labels
      test_multi_df.csv
      test_single_df.csv
  fe-train
      train_multi_df.csv
      train_single_df.csv
  fe-trial-val
      trial_val_multi_df.csv
      trial_val_single_df.csv
  test-labels
      lcp_multi_test.tsv
      lcp_single_test.tsv
  train
      lcp_multi_train.tsv
      lcp_single_train.tsv
  trial
      lcp_multi_trial.tsv
      lcp_single_trial.tsv

6 directories, 12 files
```

[9]:
```
!ls -R /content/drive/MyDrive/266-final/data/266-comp-lex-master/
```

```
/content/drive/MyDrive/266-final/data/266-comp-lex-master/:
fe-test-labels  fe-train  fe-trial-val  test-labels  train  trial

/content/drive/MyDrive/266-final/data/266-comp-lex-master/fe-test-labels:
test_multi_df.csv  test_single_df.csv

/content/drive/MyDrive/266-final/data/266-comp-lex-master/fe-train:
train_multi_df.csv  train_single_df.csv

/content/drive/MyDrive/266-final/data/266-comp-lex-master/fe-trial-val:
trial_val_multi_df.csv  trial_val_single_df.csv
```

```
/content/drive/MyDrive/266-final/data/266-comp-lex-master/test-labels:
lcp_multi_test.tsv  lcp_single_test.tsv

/content/drive/MyDrive/266-final/data/266-comp-lex-master/train:
lcp_multi_train.tsv  lcp_single_train.tsv

/content/drive/MyDrive/266-final/data/266-comp-lex-master/trial:
lcp_multi_trial.tsv  lcp_single_trial.tsv
```

[10]:
```
#@title Import Data
```

[11]:
```
# Load train data into train_*_df
train_single_df = pd.read_csv(
    os.path.join(dir_data, "train", "lcp_single_train.tsv"),
    sep = "\t",
    engine = "python",
    quoting = 3
)
train_multi_df = pd.read_csv(
    os.path.join(dir_data, "train", "lcp_multi_train.tsv"),
    sep = "\t",
    engine = "python",
    quoting = 3
)

# Load trial data into trial_val_*_df
trial_val_single_df = pd.read_csv(
    os.path.join(dir_data, "trial", "lcp_single_trial.tsv"),
    sep = "\t",
    engine = "python",
    quoting = 3
)
trial_val_multi_df = pd.read_csv(
    os.path.join(dir_data, "trial", "lcp_multi_trial.tsv"),
    sep = "\t",
    engine = "python",
    quoting = 3
)

# Load test data (with labels) into test_*_df
test_single_df = pd.read_csv(
    os.path.join(dir_data, "test-labels", "lcp_single_test.tsv"),
    sep = "\t",
    engine = "python",
    quoting = 3
)
```

```
test_multi_df = pd.read_csv(
    os.path.join(dir_data, "test-labels", "lcp_multi_test.tsv"),
    sep = "\t",
    engine = "python",
    quoting = 3
)

print("Data successfully loaded into train, trial-val, and test variables")
```

```
Data successfully loaded into train, trial-val, and test variables
```

[12]:
```
#@title EDA
```

[13]:
```
def print_dataframe_summary(df_name, df):
    # Print section header
    print(f"========== {df_name} ==========")

    # Shape and Columns
    print(f"Shape: {df.shape}")
    print(f"Columns: {list(df.columns)}\n")

    # Data Types
    print("Data Types:")
    print(df.dtypes)
    print()

    # Missing Values
    print("Missing Values (by column):")
    print(df.isna().sum())
    print()

    # 'complexity' column stats
    desc = df['complexity'].describe()  # count, mean, std, min, 25%, 50%, 75%, max
    print("'complexity' Column Stats (incl. quartiles and median):")
    print(desc)

    # Calculate frequency counts for each quartile range
    q1 = desc['25%']
    q2 = desc['50%']  # This is the median
    q3 = desc['75%']
    q_max = desc['max']

    # Note: We'll define the ranges as:
    #   <= Q1
    #   > Q1 and <= Q2
    #   > Q2 and <= Q3
```

```
    #   > Q3

    freq_q1 = np.sum(df['complexity'] <= q1)
    freq_q2 = np.sum((df['complexity'] > q1) & (df['complexity'] <= q2))
    freq_q3 = np.sum((df['complexity'] > q2) & (df['complexity'] <= q3))
    freq_q4 = np.sum(df['complexity'] > q3)

    print()
    print("Quartile Frequency Counts (tab-separated next to each quartile):")
    print(f"25%: {q1}\tCount (<= Q1): {freq_q1}")
    print(f"50% (Median): {q2}\tCount (Q1 < x <= Q2): {freq_q2}")
    print(f"75%: {q3}\tCount (Q2 < x <= Q3): {freq_q3}")
    print(f"100% (Max): {q_max}\tCount (Q3 < x <= Max): {freq_q4}")

    print("=====================================\n")

# Now we call this for each of our dataframes
print_dataframe_summary("train_single_df", train_single_df)
print_dataframe_summary("train_multi_df", train_multi_df)
print_dataframe_summary("trial_val_single_df", trial_val_single_df)
print_dataframe_summary("trial_val_multi_df", trial_val_multi_df)
print_dataframe_summary("test_single_df", test_single_df)
print_dataframe_summary("test_multi_df", test_multi_df)
```

```
========== train_single_df ==========
Shape: (7662, 5)
Columns: ['id', 'corpus', 'sentence', 'token', 'complexity']

Data Types:
id             object
corpus         object
sentence       object
token          object
complexity    float64
dtype: object

Missing Values (by column):
id            0
corpus        0
sentence      0
token         7
complexity    0
dtype: int64

'complexity' Column Stats (incl. quartiles and median):
count    7662.000000
mean        0.302288
std         0.132977
```

```
min         0.000000
25%         0.211538
50%         0.279412
75%         0.375000
max         0.861111
Name: complexity, dtype: float64

Quartile Frequency Counts (tab-separated next to each quartile):
25%: 0.2115384615384615 Count (<= Q1): 1928
50% (Median): 0.2794117647058823        Count (Q1 < x <= Q2): 1937
75%: 0.375      Count (Q2 < x <= Q3): 1984
100% (Max): 0.8611111111111112  Count (Q3 < x <= Max): 1813
======================================

========== train_multi_df ==========
Shape: (1517, 5)
Columns: ['id', 'corpus', 'sentence', 'token', 'complexity']

Data Types:
id             object
corpus         object
sentence       object
token          object
complexity    float64
dtype: object

Missing Values (by column):
id            0
corpus        0
sentence      0
token         0
complexity    0
dtype: int64

'complexity' Column Stats (incl. quartiles and median):
count    1517.000000
mean        0.418362
std         0.155536
min         0.027778
25%         0.302632
50%         0.409091
75%         0.529412
max         0.975000
Name: complexity, dtype: float64

Quartile Frequency Counts (tab-separated next to each quartile):
25%: 0.3026315789473685 Count (<= Q1): 382
50% (Median): 0.4090909090909091 Count (Q1 < x <= Q2): 377
```

```
75%: 0.5294117647058824 Count (Q2 < x <= Q3): 380
100% (Max): 0.975       Count (Q3 < x <= Max): 378
======================================

========== trial_val_single_df ==========
Shape: (421, 5)
Columns: ['id', 'subcorpus', 'sentence', 'token', 'complexity']

Data Types:
id             object
subcorpus      object
sentence       object
token          object
complexity    float64
dtype: object

Missing Values (by column):
id            0
subcorpus     0
sentence      0
token         0
complexity    0
dtype: int64

'complexity' Column Stats (incl. quartiles and median):
count    421.000000
mean       0.298631
std        0.137619
min        0.000000
25%        0.214286
50%        0.266667
75%        0.359375
max        0.875000
Name: complexity, dtype: float64

Quartile Frequency Counts (tab-separated next to each quartile):
25%: 0.2142857142857143 Count (<= Q1): 106
50% (Median): 0.2666666666666667        Count (Q1 < x <= Q2): 107
75%: 0.359375   Count (Q2 < x <= Q3): 103
100% (Max): 0.875       Count (Q3 < x <= Max): 105
======================================

========== trial_val_multi_df ==========
Shape: (99, 5)
Columns: ['id', 'subcorpus', 'sentence', 'token', 'complexity']

Data Types:
id             object
```

```
subcorpus      object
sentence       object
token          object
complexity    float64
dtype: object

Missing Values (by column):
id            0
subcorpus     0
sentence      0
token         0
complexity    0
dtype: int64

'complexity' Column Stats (incl. quartiles and median):
count    99.000000
mean      0.417961
std       0.153752
min       0.000000
25%       0.309028
50%       0.421875
75%       0.513932
max       0.825000
Name: complexity, dtype: float64

Quartile Frequency Counts (tab-separated next to each quartile):
25%: 0.3090277777777778 Count (<= Q1): 25
50% (Median): 0.421875  Count (Q1 < x <= Q2): 25
75%: 0.5139318885448916 Count (Q2 < x <= Q3): 24
100% (Max): 0.825       Count (Q3 < x <= Max): 25
======================================

========== test_single_df ==========
Shape: (917, 5)
Columns: ['id', 'corpus', 'sentence', 'token', 'complexity']

Data Types:
id             object
corpus         object
sentence       object
token          object
complexity    float64
dtype: object

Missing Values (by column):
id            0
corpus        0
sentence      0
```

```
token         0
complexity    0
dtype: int64

'complexity' Column Stats (incl. quartiles and median):
count    917.000000
mean       0.296362
std        0.127290
min        0.000000
25%        0.214286
50%        0.276316
75%        0.357143
max        0.777778
Name: complexity, dtype: float64

Quartile Frequency Counts (tab-separated next to each quartile):
25%: 0.2142857142857143 Count (<= Q1): 237
50% (Median): 0.2763157894736842        Count (Q1 < x <= Q2): 224
75%: 0.3571428571428571 Count (Q2 < x <= Q3): 229
100% (Max): 0.7777777777777777  Count (Q3 < x <= Max): 227
====================================

========== test_multi_df ==========
Shape: (184, 5)
Columns: ['id', 'corpus', 'sentence', 'token', 'complexity']

Data Types:
id             object
corpus         object
sentence       object
token          object
complexity    float64
dtype: object

Missing Values (by column):
id            0
corpus        0
sentence      0
token         0
complexity    0
dtype: int64

'complexity' Column Stats (incl. quartiles and median):
count    184.000000
mean       0.422312
std        0.155785
min        0.000000
25%        0.316667
```

```
50%        0.428571
75%        0.527778
max        0.800000
Name: complexity, dtype: float64

Quartile Frequency Counts (tab-separated next to each quartile):
25%: 0.3166666666666666 Count (<= Q1): 47
50% (Median): 0.4285714285714286        Count (Q1 < x <= Q2): 46
75%: 0.5277777777777778 Count (Q2 < x <= Q3): 46
100% (Max): 0.8 Count (Q3 < x <= Max): 45
====================================
```

[14]:
```
print(train_single_df.head())
```

```
                               id corpus  \
0  3ZLW647WALVGE8EBR50EGUBPU4P32A  bible
1  34R0BODSP1ZBN3DVY8J8XSIY551E5C  bible
2  3S1WOPCJFGTJU2SGNAN2Y213N6WJE3  bible
3  3BFNCI9LYKQN09BHXHH9CLSX5KP738  bible
4  3G5RUKN2EC3YIWSKUXZ8ZVH95R49N2  bible

                                        sentence     token  complexity
0  Behold, there came up out of the river seven c…     river    0.000000
1  I am a fellow bondservant with you and with yo…  brothers    0.000000
2  The man, the lord of the land, said to us, 'By…  brothers    0.050000
3  Shimei had sixteen sons and six daughters; but…  brothers    0.150000
4               "He has put my brothers far from me.  brothers    0.263889
```

[15]:
```
print(train_multi_df.head())
```

```
                               id corpus  \
0  3S37Y8CWI80N8KVM53U4E6JKCDC4WE  bible
1  3WGCNLZJKF877FYC1Q6COKNWTDWD11  bible
2  3UOMW19E6D6WQ5TH2HDD74IVKTP5CB  bible
3  36JW4WBR06KF9AXMUL4N476OMF8FHD  bible
4  3HRWUH63QU2FH9Q8R7MRNFC7JX2N5A  bible

                                        sentence            token  \
0  but the seventh day is a Sabbath to Yahweh you…      seventh day
1  But let each man test his own work, and then h…         own work
2  To him who by understanding made the heavens; …  loving kindness
3  Remember to me, my God, this also, and spare m…  loving kindness
4  Because your loving kindness is better than li…  loving kindness

   complexity
0    0.027778
1    0.050000
2    0.050000
```

```
3    0.050000
4    0.075000
```

[16]:
```
#@title Data Engineering
```

[17]:
```
# Assuming you have already loaded the DataFrames:
# train_single_df, train_multi_df, trial_val_single_df, trial_val_multi_df,␣
 ↪test_single_df, test_multi_df

def print_distinct_values(df, column_name):
    """Prints the distinct values of a specified column in a DataFrame."""
    distinct_values = df[column_name].unique()
    print(f"Distinct values in '{column_name}' column:")
    for value in distinct_values:
        print(value)
    print("-" * 30)  # Separator

# Print distinct values for each DataFrame
print_distinct_values(train_single_df, "corpus")
print_distinct_values(train_multi_df, "corpus")
print_distinct_values(trial_val_single_df, "subcorpus")
print_distinct_values(trial_val_multi_df, "subcorpus")
print_distinct_values(test_single_df, "corpus")
print_distinct_values(test_multi_df, "corpus")
```

```
Distinct values in 'corpus' column:
bible
biomed
europarl
------------------------------
Distinct values in 'corpus' column:
bible
biomed
europarl
------------------------------
Distinct values in 'subcorpus' column:
bible
biomed
europarl
------------------------------
Distinct values in 'subcorpus' column:
bible
biomed
europarl
------------------------------
Distinct values in 'corpus' column:
bible
biomed
```

```
europarl
------------------------------
Distinct values in 'corpus' column:
bible
biomed
europarl
------------------------------
```

0.1 standardize column headers: convert trial_val header from 'subcorpus' to 'corpus'

[18]:
```
# Rename the 'subcorpus' column to 'corpus'
trial_val_single_df = trial_val_single_df.rename(columns={'subcorpus':␣
 ↪'corpus'})
trial_val_multi_df = trial_val_multi_df.rename(columns={'subcorpus': 'corpus'})

# Verify the change (optional)
print(trial_val_single_df.columns)
print(trial_val_multi_df.columns)
```

```
Index(['id', 'corpus', 'sentence', 'token', 'complexity'], dtype='object')
Index(['id', 'corpus', 'sentence', 'token', 'complexity'], dtype='object')
```

[19]:
```
dataframes = [train_single_df, train_multi_df, trial_val_single_df,␣
 ↪trial_val_multi_df, test_single_df, test_multi_df]

# Get the headers (column names) of the first DataFrame as a reference
reference_headers = list(dataframes[0].columns)

# Loop through the remaining DataFrames and compare headers
all_headers_match = True
for df in dataframes[1:]:
    if list(df.columns) != reference_headers:
        all_headers_match = False
        print(f"Headers do not match for DataFrame: {df.head(0)}")  # Print␣
 ↪which DataFrame has different headers
        break  # Exit the loop if a mismatch is found

# Print the result
if all_headers_match:
    print("All DataFrames have matching headers.")
else:
    print("Headers do not match for all DataFrames.")
```

```
All DataFrames have matching headers.
```

0.2 Interrogate Span Length by Corpus Value by Data Split

[20]:
```
tokenizer = RegexpTokenizer(r'\w+')

def analyze_sentence_spans_by_corpus_and_quartile(dfs_dict):
    """
    Analyze sentence spans (length metrics) grouped by corpus and complexity␣
 ↪quartile
    for multiple dataframes.
    """
    results = []

    for df_name, df in dfs_dict.items():
        print(f"Processing {df_name}...")

        q1 = df['complexity'].quantile(0.25)
        q2 = df['complexity'].quantile(0.50)
        q3 = df['complexity'].quantile(0.75)

        def get_quartile(x):
            if x <= q1:
                return 'Q1'
            elif x <= q2:
                return 'Q2'
            elif x <= q3:
                return 'Q3'
            else:
                return 'Q4'

        df = df.copy()
        df['quartile'] = df['complexity'].apply(get_quartile)

        def compute_span_metrics(sentence):
            if pd.isna(sentence):
                return pd.Series({'word_count': 0, 'char_count': 0,␣
 ↪'avg_word_len': 0})

            words = tokenizer.tokenize(sentence)
            word_count = len(words)
            char_count = len(sentence)
            avg_word_len = np.mean([len(word) for word in words]) if word_count␣
 ↪> 0 else 0
            return pd.Series({'word_count': word_count, 'char_count':␣
 ↪char_count, 'avg_word_len': avg_word_len})

        span_metrics = df['sentence'].apply(compute_span_metrics)
        df = pd.concat([df, span_metrics], axis=1)
```

```
        corpus_col = 'corpus' if 'corpus' in df.columns else 'subcorpus'

        for corpus_name, corpus_df in df.groupby(corpus_col):
            for quartile, quartile_df in corpus_df.groupby('quartile'):
                complexity_range = f"{quartile_df['complexity'].min():.
↪3f}-{quartile_df['complexity'].max():.3f}"
                stats = {
                    'Dataframe': df_name,
                    'Corpus': corpus_name,
                    'Quartile': quartile,
                    'Complexity Range': complexity_range,
                    'Count': len(quartile_df),
                    'Avg Words': quartile_df['word_count'].mean(),
                    'Median Words': quartile_df['word_count'].median(),
                    'Min Words': quartile_df['word_count'].min(),
                    'Max Words': quartile_df['word_count'].max(),
                    'Std Words': quartile_df['word_count'].std(),
                    'Avg Chars': quartile_df['char_count'].mean(),
                    'Avg Word Len': quartile_df['avg_word_len'].mean()
                }
                results.append(stats)

    results_df = pd.DataFrame(results)
    results_df = results_df.sort_values(['Dataframe', 'Corpus', 'Quartile'])

    return results_df

dfs = {
    'train_single_df': train_single_df,
    'train_multi_df': train_multi_df,
    'trial_val_single_df': trial_val_single_df,
    'trial_val_multi_df': trial_val_multi_df,
    'test_single_df': test_single_df,
    'test_multi_df': test_multi_df
}

span_analysis = analyze_sentence_spans_by_corpus_and_quartile(dfs)

pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)
pd.set_option('display.width', 1000)
display(span_analysis)

results_path = os.path.join(dir_results, 'sentence_span_analysis.csv')
span_analysis.to_csv(results_path, index=False)
print(f"Analysis saved to: {results_path}")
```

```
Processing train_single_df…
Processing train_multi_df…
Processing trial_val_single_df…
Processing trial_val_multi_df…
Processing test_single_df…
Processing test_multi_df…
```

	Dataframe	Corpus	Quartile	Complexity Range	Count	Avg Words	Median Words	Min Words	Max Words	Std Words	Avg Chars	Avg Word Len
60	test_multi_df	bible	Q1	0.025-0.317	26	23.076923	22.0	4.0	48.0	11.831900	118.653846	4.128898
61	test_multi_df	bible	Q2	0.325-0.417	11	20.545455	17.0	7.0	47.0	12.917923	109.545455	4.209752
62	test_multi_df	bible	Q3	0.432-0.528	18	21.111111	21.5	4.0	43.0	10.889222	112.777778	4.474206
63	test_multi_df	bible	Q4	0.542-0.694	11	22.363636	20.0	7.0	51.0	11.935432	126.181818	4.605062
64	test_multi_df	biomed	Q1	0.000-0.312	11	29.818182	29.0	17.0	47.0	8.388304	195.727273	5.491145
65	test_multi_df	biomed	Q2	0.324-0.417	11	27.090909	24.0	9.0	47.0	11.449494	171.818182	5.436237
66	test_multi_df	biomed	Q3	0.456-0.528	10	26.900000	26.5	10.0	49.0	10.712921	177.500000	5.497409
67	test_multi_df	biomed	Q4	0.562-0.800	21	32.285714	34.0	14.0	56.0	13.598319	209.285714	5.460101
68	test_multi_df	europarl	Q1	0.214-0.303	10	24.700000	24.5	7.0	56.0	14.189589	146.900000	5.049688
69	test_multi_df	europarl	Q2	0.321-0.429	24	27.833333	27.0	9.0	73.0	15.352855	172.291667	5.269610
70	test_multi_df	europarl	Q3	0.432-0.516	18	32.944444	32.0	6.0	68.0	19.129504	209.888889	5.512245
71	test_multi_df	europarl	Q4	0.531-0.562	13	39.000000	36.0	6.0	95.0	29.631065	237.076923	5.100616
48	test_single_df	bible	Q1	0.000-0.214	79	22.835443	22.0	7.0	49.0	10.602891	116.797468	4.031532
49	test_single_df	bible	Q2	0.217-0.276	68	24.176471	21.0	2.0	77.0	14.393138	125.955882	4.167352
50	test_single_df	bible	Q3	0.278-0.353	67	22.388060	20.0	4.0	63.0	11.306950	119.731343	4.254090
51	test_single_df	bible	Q4	0.359-0.732	69	20.579710	19.0	1.0	55.0	11.264736	110.550725	4.337010
52	test_single_df	biomed	Q1	0.000-0.214	75	27.080000	25.0	10.0	84.0	12.025603	172.893333	5.271985
53	test_single_df	biomed	Q2	0.217-0.275	58	30.275862	26.0	10.0	83.0	15.856587	197.775862	5.434573
54	test_single_df	biomed	Q3	0.278-0.357	66	29.833333	29.0	13.0	85.0	11.754650	191.863636	5.334048

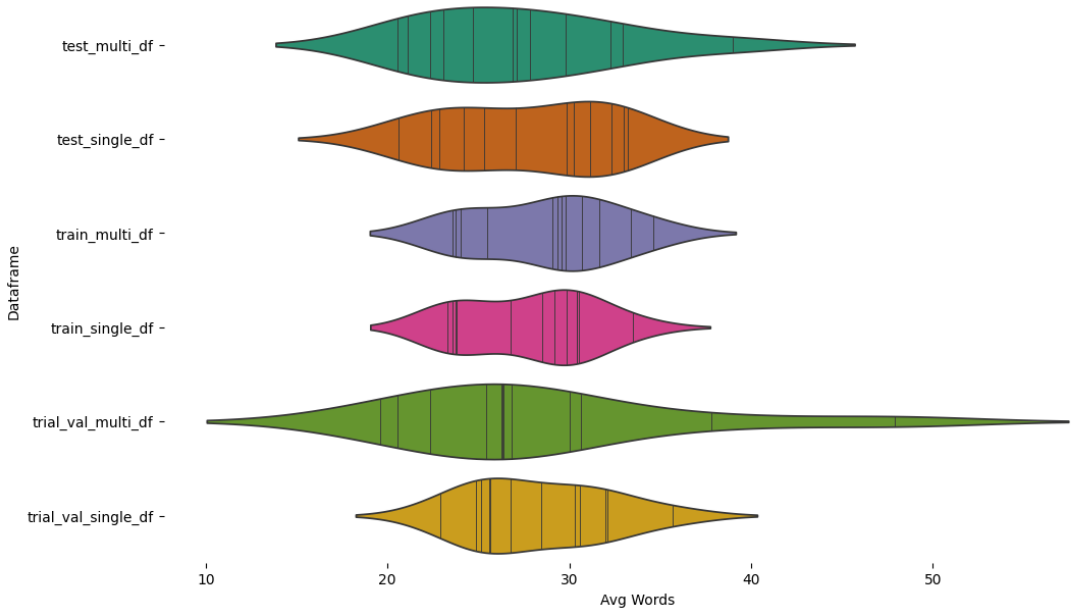
```
55      test_single_df    biomed       Q4      0.359-0.778     90   31.144444
  ↪     30.0       14.0       83.0  12.089146  203.055556       5.393138
56      test_single_df  europarl       Q1      0.000-0.214     83   25.337349
  ↪     21.0        3.0       82.0  16.032191  151.891566       5.044222
57      test_single_df  europarl       Q2      0.217-0.276     98   32.326531
  ↪     30.0        1.0       97.0  18.707061  195.653061       5.062296
58      test_single_df  europarl       Q3      0.278-0.357     96   33.000000
  ↪     30.0        3.0      141.0  21.404377  201.760417       5.124551
59      test_single_df  europarl       Q4      0.361-0.583     68   33.235294
  ↪     29.0        1.0      130.0  20.440023  206.514706       5.164123
12      train_multi_df     bible       Q1      0.028-0.300    163   23.588957
  ↪     22.0        3.0       67.0  12.429421  124.834356       4.232989
13      train_multi_df     bible       Q2      0.304-0.409    132   24.053030
  ↪     22.0        6.0       65.0  11.738444  129.575758       4.302615
14      train_multi_df     bible       Q3      0.411-0.529    131   23.770992
  ↪     23.0        4.0       50.0  11.158691  127.389313       4.324088
15      train_multi_df     bible       Q4      0.533-0.778     79   25.481013
  ↪     24.0        3.0       81.0  13.490605  139.240506       4.486716
16      train_multi_df    biomed       Q1      0.028-0.303     87   29.091954
  ↪     28.0        9.0       77.0  11.882792  185.954023       5.276290
17      train_multi_df    biomed       Q2      0.304-0.408     74   30.716216
  ↪     28.0       11.0       85.0  13.521693  195.864865       5.370313
18      train_multi_df    biomed       Q3      0.411-0.529    111   29.783784
  ↪     29.0        8.0       61.0  10.912383  193.855856       5.430133
19      train_multi_df    biomed       Q4      0.531-0.975    242   29.595041
  ↪     28.0       10.0       75.0  12.040443  194.995868       5.534629
20      train_multi_df  europarl       Q1      0.118-0.303    132   29.363636
  ↪     27.0        3.0      101.0  17.874146  176.553030       5.002618
21      train_multi_df  europarl       Q2      0.304-0.409    171   31.654971
  ↪     28.0        3.0      108.0  19.099221  195.152047       5.176834
22      train_multi_df  europarl       Q3      0.411-0.529    138   33.398551
  ↪     30.0        7.0      101.0  18.992715  208.304348       5.286607
23      train_multi_df  europarl       Q4      0.533-0.750     57   34.596491
  ↪     31.0        6.0       96.0  20.318763  218.350877       5.345891
0     train_single_df     bible       Q1      0.000-0.212    701   23.275321
  ↪     22.0        4.0       61.0  11.760701  121.607703       4.126789
1     train_single_df     bible       Q2      0.212-0.279    640   23.753125
  ↪     22.0        3.0       60.0  11.577932  124.576562       4.148961
2     train_single_df     bible       Q3      0.281-0.375    624   23.823718
  ↪     22.0        3.0       70.0  11.958906  126.230769       4.208102
3     train_single_df     bible       Q4      0.380-0.861    609   23.577997
  ↪     21.0        3.0       69.0  12.461688  126.518883       4.295608
4     train_single_df    biomed       Q1      0.000-0.212    586   28.534130
  ↪     27.0        2.0       85.0  12.115387  182.011945       5.319754
5     train_single_df    biomed       Q2      0.212-0.279    583   30.435678
  ↪     29.0        7.0       92.0  11.872558  193.789022       5.285758
```

```
6       train_single_df    biomed       Q3      0.281-0.375    659  29.860395
  ↪      28.0        4.0       77.0  11.591263  191.050076      5.328161
7       train_single_df    biomed       Q4      0.381-0.861    748  29.176471
  ↪      28.0        3.0       85.0  12.246613  186.909091      5.298112
8       train_single_df  europarl       Q1      0.025-0.212    641  26.761310
  ↪      24.0        2.0      107.0  15.230853  159.180967      4.942557
9       train_single_df  europarl       Q2      0.212-0.279    714  30.420168
  ↪      27.0        1.0      129.0  18.383783  183.093838      4.995672
10      train_single_df  europarl       Q3      0.281-0.375    701  30.523538
  ↪      28.0        1.0      122.0  18.163026  185.840228      5.114587
11      train_single_df  europarl       Q4      0.381-0.775    456  33.528509
  ↪      31.0        2.0      235.0  21.704693  203.592105      5.054701
36   trial_val_multi_df     bible       Q1      0.000-0.292     11  26.272727
  ↪      21.0       13.0       64.0  13.950562  141.363636      4.282457
37   trial_val_multi_df     bible       Q2      0.333-0.400      7  20.571429
  ↪      23.0        5.0       28.0   7.412987  110.857143      4.279406
38   trial_val_multi_df     bible       Q3      0.425-0.500      5  19.600000
  ↪      19.0        9.0       32.0   8.905055  109.200000      4.431391
39   trial_val_multi_df     bible       Q4      0.525-0.661      6  22.333333
  ↪      20.5        9.0       44.0  12.242004  117.833333      4.178525
40   trial_val_multi_df    biomed       Q1      0.083-0.303      6  26.833333
  ↪      25.0       15.0       49.0  11.771434  159.166667      4.899969
41   trial_val_multi_df    biomed       Q2      0.317-0.422      7  25.428571
  ↪      21.0       15.0       48.0  11.588171  156.000000      5.194383
42   trial_val_multi_df    biomed       Q3      0.438-0.513      6  37.833333
  ↪      39.5       26.0       44.0   6.675827  247.500000      5.438593
43   trial_val_multi_df    biomed       Q4      0.537-0.825     14  30.642857
  ↪      29.5       17.0       43.0   9.849695  211.428571      5.730623
44   trial_val_multi_df  europarl       Q1      0.176-0.306      8  30.000000
  ↪      25.5        4.0       64.0  20.361027  186.750000      5.306837
45   trial_val_multi_df  europarl       Q2      0.312-0.412     11  47.909091
  ↪      46.0       24.0       78.0  18.651834  296.909091      5.058375
46   trial_val_multi_df  europarl       Q3      0.432-0.500     13  26.307692
  ↪      26.0        5.0       66.0  18.167666  166.153846      5.263847
47   trial_val_multi_df  europarl       Q4      0.515-0.714      5  26.400000
  ↪      15.0        6.0       66.0  24.316661  164.600000      4.998182
24  trial_val_single_df     bible       Q1      0.000-0.214     52  26.750000
  ↪      26.0        5.0       73.0  15.530962  137.230769      4.071006
25  trial_val_single_df     bible       Q2      0.217-0.266     38  24.868421
  ↪      23.0        7.0       50.0  10.768249  131.236842      4.195550
26  trial_val_single_df     bible       Q3      0.268-0.355     26  22.884615
  ↪      20.5        5.0       44.0   9.961233  121.269231      4.312026
27  trial_val_single_df     bible       Q4      0.361-0.633     27  25.666667
  ↪      23.0        6.0       49.0  12.554497  137.555556      4.212685
28  trial_val_single_df    biomed       Q1      0.028-0.214     21  25.571429
  ↪      21.0       13.0       65.0  11.543706  163.904762      5.305404
```

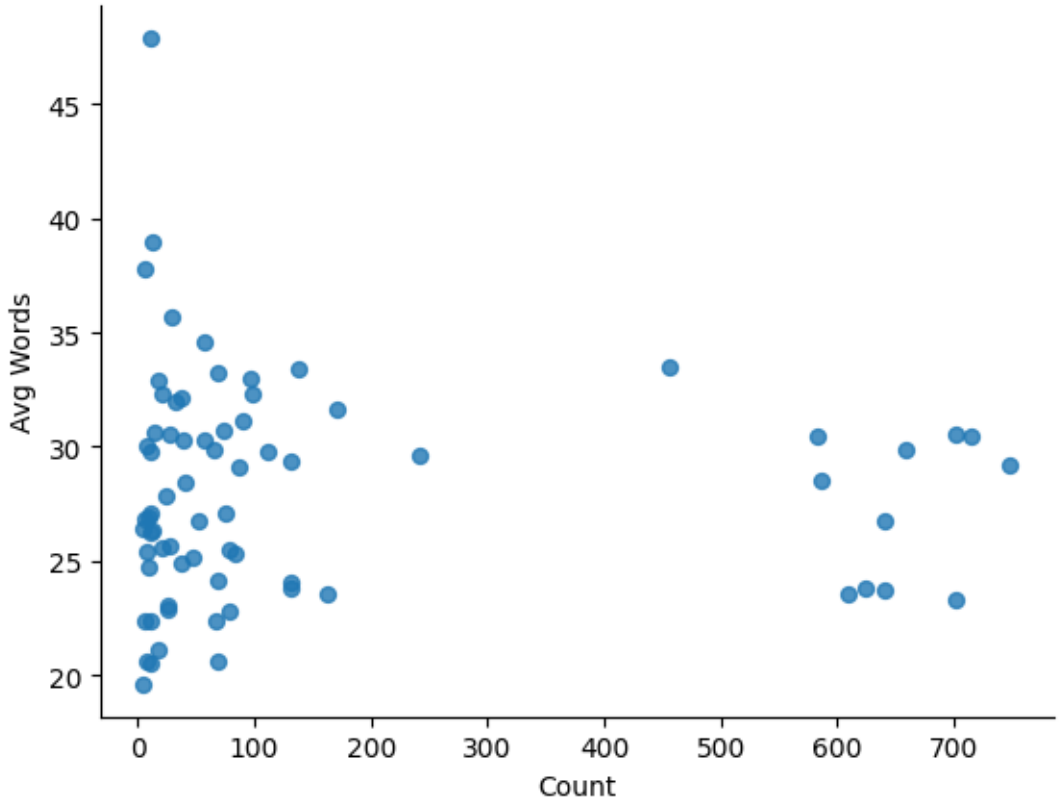
```
29  trial_val_single_df    biomed       Q2      0.217-0.267     28  30.571429
 ↪      27.5       11.0       57.0  12.099674  198.142857      5.315287
30  trial_val_single_df    biomed       Q3      0.268-0.359     38  32.105263
 ↪      29.0       11.0       61.0  12.710476  206.947368      5.364934
31  trial_val_single_df    biomed       Q4      0.364-0.875     48  25.145833
 ↪      25.5        6.0       56.0  11.721937  163.979167      5.439709
32  trial_val_single_df  europarl       Q1      0.050-0.214     33  31.969697
 ↪      28.0        5.0       81.0  20.356947  185.969697      4.799024
33  trial_val_single_df  europarl       Q2      0.217-0.267     41  28.463415
 ↪      28.0        4.0       71.0  15.386841  172.780488      4.997706
34  trial_val_single_df  europarl       Q3      0.268-0.359     39  30.282051
 ↪      28.0        3.0       99.0  20.040681  184.358974      5.086945
35  trial_val_single_df  europarl       Q4      0.367-0.605     30  35.700000
 ↪      30.5        5.0       77.0  20.142852  215.400000      4.910759

Analysis saved to:
/content/drive/MyDrive/266-final/results/sentence_span_analysis.csv
```

[21]:
```
from matplotlib import pyplot as plt
import seaborn as sns
figsize = (12, 1.2 * len(span_analysis['Dataframe'].unique()))
plt.figure(figsize=figsize)
sns.violinplot(span_analysis, x='Avg Words', y='Dataframe', inner='stick',
 ↪palette='Dark2')
sns.despine(top=True, right=True, bottom=True, left=True)
```

```
<ipython-input-21-00a8ad5642c1>:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same
effect.

  sns.violinplot(span_analysis, x='Avg Words', y='Dataframe', inner='stick',
palette='Dark2')
```

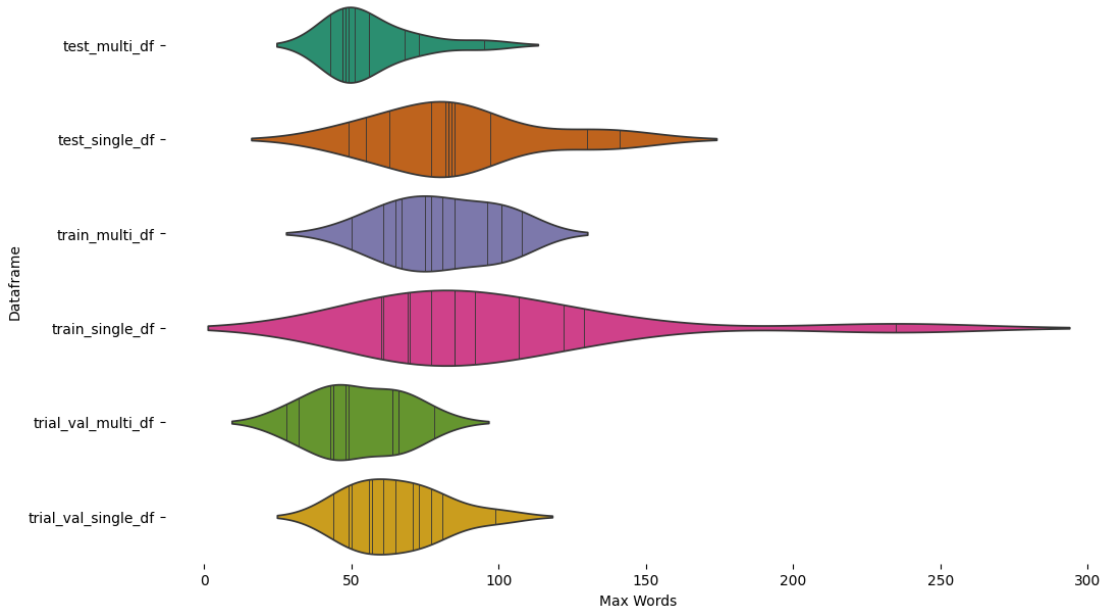


[22]:
```
from matplotlib import pyplot as plt
span_analysis.plot(kind='scatter', x='Count', y='Avg Words', s=32, alpha=.8)
plt.gca().spines[['top', 'right',]].set_visible(False)
```



[23]:
```
from matplotlib import pyplot as plt
import seaborn as sns
figsize = (12, 1.2 * len(span_analysis['Dataframe'].unique()))
plt.figure(figsize=figsize)
sns.violinplot(span_analysis, x='Max Words', y='Dataframe', inner='stick',␣
↪palette='Dark2')
sns.despine(top=True, right=True, bottom=True, left=True)
```

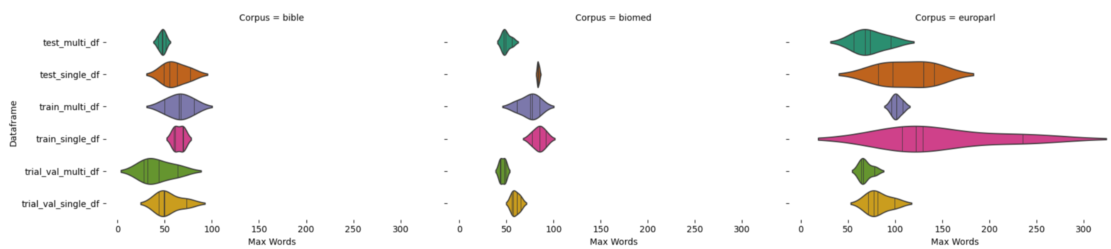
```
<ipython-input-23-01bf0c89d620>:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same
effect.

  sns.violinplot(span_analysis, x='Max Words', y='Dataframe', inner='stick',
palette='Dark2')
```



[24]:
```
g = sns.FacetGrid(span_analysis, col="Corpus", col_wrap=3, height=4, aspect=1.5)
g.map(sns.violinplot, "Max Words", "Dataframe", inner='stick', palette='Dark2')
g.despine(top=True, right=True, bottom=True, left=True)
plt.tight_layout()
plt.show()
```

```
/usr/local/lib/python3.11/dist-packages/seaborn/axisgrid.py:718: UserWarning:
Using the violinplot function without specifying `order` is likely to produce an
```

```
incorrect plot.
  warnings.warn(warning)
/usr/local/lib/python3.11/dist-packages/seaborn/axisgrid.py:854: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same
effect.

  func(*plot_args, **plot_kwargs)
/usr/local/lib/python3.11/dist-packages/seaborn/axisgrid.py:854: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same
effect.

  func(*plot_args, **plot_kwargs)
/usr/local/lib/python3.11/dist-packages/seaborn/axisgrid.py:854: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same
effect.

  func(*plot_args, **plot_kwargs)
```



- decision: no modifications to sentence spans will be applied, except for Contraction standardization

0.3 Normalize / Eliminate Contractions

[25]:
```
def expand_contractions_in_df(df):
    """
    1) Creates a new column 'sentence_no_contractions' by expanding any␣
↪contractions.
    2) Identifies rows where a contraction was actually expanded (the text␣
↪changed).
    3) Returns the updated DataFrame and a grouped subset of rows for printing␣
↪examples.
```

```
    """
    df = df.copy()
    df['sentence_no_contractions'] = df['sentence'].apply(
        lambda s: contractions.fix(s) if pd.notna(s) else s
    )

    df['contraction_expanded'] = df.apply(
        lambda row: row['sentence'] != row['sentence_no_contractions'], axis=1
    )

    results_by_corpus = {}
    for corpus_val, group in df.groupby('corpus'):
        changed_rows = group[group['contraction_expanded']]
        first_three = changed_rows.head(3)
        results_by_corpus[corpus_val] = first_three
    return df, results_by_corpus


dataframes_info = [
    ("train_single_df", train_single_df),
    ("train_multi_df", train_multi_df),
    ("trial_val_single_df", trial_val_single_df),
    ("trial_val_multi_df", trial_val_multi_df),
    ("test_single_df", test_single_df),
    ("test_multi_df", test_multi_df),
]

for df_name, df in dataframes_info:
    updated_df, corpus_examples = expand_contractions_in_df(df)
    globals()[df_name] = updated_df

    print(f"\n{'='*60}")
    print(f"DataFrame: {df_name}")
    print(f"{'='*60}")

    for corpus_val in sorted(corpus_examples.keys()):
        subset = corpus_examples[corpus_val]
        if len(subset) == 0:
            continue
        print(f"\n  Corpus: {corpus_val}")
        print("    -- BEFORE --")
        for _, row in subset.iterrows():
            print(f"       {row['sentence']}")
        print("    -- AFTER  --")
        for _, row in subset.iterrows():
            print(f"       {row['sentence_no_contractions']}")
```

```
============================================================
DataFrame: train_single_df
============================================================

  Corpus: bible
    -- BEFORE --
       Shimei had sixteen sons and six daughters; but his brothers didn't have
many children, neither did all their family multiply like the children of Judah.
       When his speech is charming, don't believe him; for there are seven
abominations in his heart.
       Jesus said, "Father, forgive them, for they don't know what they are
doing."
    -- AFTER  --
       Shimei had sixteen sons and six daughters; but his brothers did not have
many children, neither did all their family multiply like the children of Judah.
       When his speech is charming, do not believe him; for there are seven
abominations in his heart.
       Jesus said, "Father, forgive them, for they do not know what they are
doing."

  Corpus: biomed
    -- BEFORE --
       Although missense mutation of ITPR1 had previously been ruled out [2] and
the mode of inheritance was inconsistent with that seen in the Itpr1Δ18 and
Itpr1opt mice, the phenotypic presence of ataxia in the mice led us to reexamine
this candidate gene as a possible cause of SCA15.
       Human germline mutations in APC cause FAP [4,5], which is characterized
by hundreds of adenomatous colorectal polyps, with an almost inevitable
progression to colorectal cancer in the third and fourth decades of life.
       Null mutations in Bmpr1a cause early embryonic lethality, with defects in
gastrulation similar to those seen in mice with mutations in Bmp4 (Mishina et
al. 1995; Winnier et al. 1995).
    -- AFTER  --
       Although missense mutation of ITPR1 had previously been ruled out [2] and
the mode of inheritance was inconsistent with that seen in the Itpr1Δ18 and
Itpr1opt mice, the phenotypic presence of ataxia in the mice led us to reexamine
this candidate gene as a possible because of SCA15.
       Human germline mutations in APC because FAP [4,5], which is characterized
by hundreds of adenomatous colorectal polyps, with an almost inevitable
progression to colorectal cancer in the third and fourth decades of life.
       Null mutations in Bmpr1a because early embryonic lethality, with defects
in gastrulation similar to those seen in mice with mutations in Bmp4 (Mishina et
al. 1995; Winnier et al. 1995).

  Corpus: europarl
    -- BEFORE --
       At the same time, you will also have an important role in winning over
the general public of the Member States to the cause of enlargement, of
```

enlargement based on conditionality.
       the recommendation for second reading from the Committee on Transport and Tourism on the common position adopted by the Council with a view to the adoption of a Regulation of the European Parliament and of the Council establishing common rules concerning the conditions to be complied with to pursue the occupation of road transport operator and repealing Council Directive 96/26/EC (11783/1/2008 - C6-0015/2009 - (Rapporteur: Silvia-Adriana Ţicău), and
       Yet, although credit rating agencies were not the main cause of the recent financial crisis, they did have a harmful influence.
    -- AFTER --
       At the same time, you will also have an important role in winning over the general public of the Member States to the because of enlargement, of enlargement based on conditionality.
       the recommendation for second reading from the Committee on Transport and Tourism on the common position adopted by the Council with a view to the adoption of a Regulation of the European Parliament and of the Council establishing common rules concerning the conditions to be complied with to pursue the occupation of road transport operator and repealing Council Directive 96/26/EC (11783/1/2008 - C6-0015/2009 - (Rapporteur: Silvia-Adriana Ţicăyou), and
       Yet, although credit rating agencies were not the main because of the recent financial crisis, they did have a harmful influence.

==
DataFrame: train_multi_df
==

  Corpus: bible
    -- BEFORE --
       Jahath was the chief, and Zizah the second: but Jeush and Beriah didn't have many sons; therefore they became a fathers' house in one reckoning.
       But Yahweh said to Samuel, "Don't look on his face, or on the height of his stature; because I have rejected him: for I see not as man sees; for man looks at the outward appearance, but Yahweh looks at the heart."
       Because indeed a notable miracle has been done through them, as can be plainly seen by all who dwell in Jerusalem, and we can't deny it.
    -- AFTER --
       Jahath was the chief, and Zizah the second: but Jeush and Beriah did not have many sons; therefore they became a fathers' house in one reckoning.
       But Yahweh said to Samuel, "Do not look on his face, or on the height of his stature; because I have rejected him: for I see not as man sees; for man looks at the outward appearance, but Yahweh looks at the heart."
       Because indeed a notable miracle has been done through them, as can be plainly seen by all who dwell in Jerusalem, and we cannot deny it.

  Corpus: biomed
    -- BEFORE --
       The aim in the present study was to determine the location of pendrin and

the cause of deafness in Slc26a4-/- mice.
       These characteristics should make RMCE-ASAP a robust and general technology for analysis of mammalian genes under conditions that preserve normal control mechanisms in different tissues.
       It was also demonstrated that mutations leading to abolishment of the enzymatic activity of CLN2 were the direct cause of a fatal inherited neurodegenerative disease, classical late-infantile neuronal ceroid lipofuscinosis [2].
    -- AFTER --
       The aim in the present study was to determine the location of pendrin and the because of deafness in Slc26a4-/- mice.
       These characteristics should make RMCE-AS SOON AS POSSIBLE a robust and general technology for analysis of mammalian genes under conditions that preserve normal control mechanisms in different tissues.
       It was also demonstrated that mutations leading to abolishment of the enzymatic activity of CLN2 were the direct because of a fatal inherited neurodegenerative disease, classical late-infantile neuronal ceroid lipofuscinosis [2].

  Corpus: europarl
    -- BEFORE --
       Account must also be taken of the costs to health, the environment and the climate of the fact that vehicles emit different types of particles and that, in burning fossil fuels, they cause increased pollution and thus more global warming.
       However, this unequal trade relationship is not the only cause for concern; another is the case of unsafe products coming from China.
       (IT) Madam President, ladies and gentlemen, the oral amendment that our Group is proposing involves replacing the words 'all forms of glorifying' by the word 'apology'.
    -- AFTER --
       Account must also be taken of the costs to health, the environment and the climate of the fact that vehicles emit different types of particles and that, in burning fossil fuels, they because increased pollution and thus more global warming.
       However, this unequal trade relationship is not the only because for concern; another is the case of unsafe products coming from China.
       (IT) Madam President, ladies and gentlemen, the oral amendment that our Group is proposing involves replacing the words forms of glorifying' by the word 'apology'.

==
DataFrame: trial_val_single_df
==

  Corpus: bible
    -- BEFORE --
       Don't curse the king, no, not in your thoughts; and don't curse the rich

in your bedroom: for a bird of the sky may carry your voice, and that which has wings may tell the matter.
       The young man didn't wait to do this thing, because he had delight in Jacob's daughter, and he was honored above all the house of his father.
       If the axe is blunt, and one doesn't sharpen the edge, then he must use more strength; but skill brings success.
    -- AFTER --
       Do not curse the king, no, not in your thoughts; and do not curse the rich in your bedroom: for a bird of the sky may carry your voice, and that which has wings may tell the matter.
       The young man did not wait to do this thing, because he had delight in Jacob's daughter, and he was honored above all the house of his father.
       If the axe is blunt, and one does not sharpen the edge, then he must use more strength; but skill brings success.

  Corpus: biomed
    -- BEFORE --
       For example, the non-BC individual and BC individual groups are not perfectly matched with respect to age, gender or smoking history (Table 1) and each of these factors could contribute to the observed difference in correlation between groups.
       EM and ER conducted transmission electron microscopy.
    -- AFTER --
       For example, the non-BECAUSE individual and BECAUSE individual groups are not perfectly matched with respect to age, gender or smoking history (Table 1) and each of these factors could contribute to the observed difference in correlation between groups.
       THEM and ER conducted transmission electron microscopy.

  Corpus: europarl
    -- BEFORE --
       With their help, John has sought to shed light on what has been a very murky area, and to bring clarity where uncertainty prevailed before, based consistently on the twin principles that the patient must always come first and that patient choice should be determined by needs and not by means.
    -- AFTER --
       With their help, John has sought to she would light on what has been a very murky area, and to bring clarity where uncertainty prevailed before, based consistently on the twin principles that the patient must always come first and that patient choice should be determined by needs and not by means.

==
DataFrame: trial_val_multi_df
==

==
DataFrame: test_single_df
==

Corpus: bible
-- BEFORE --
the ten sons of Haman the son of Hammedatha, the Jew's enemy, but they didn't lay their hand on the plunder.

Hezekiah listened to them, and showed them all the house of his precious things, the silver, and the gold, and the spices, and the precious oil, and the house of his armor, and all that was found in his treasures: there was nothing in his house, nor in all his dominion, that Hezekiah didn't show them.

Of Manasseh also there fell away some to David, when he came with the Philistines against Saul to battle; but they didn't help them; for the lords of the Philistines sent him away after consultation, saying, "He will fall away to his master Saul to the jeopardy of our heads."
-- AFTER --
the ten sons of Haman the son of Hammedatha, the Jew's enemy, but they did not lay their hand on the plunder.

Hezekiah listened to them, and showed them all the house of his precious things, the silver, and the gold, and the spices, and the precious oil, and the house of his armor, and all that was found in his treasures: there was nothing in his house, nor in all his dominion, that Hezekiah did not show them.

Of Manasseh also there fell away some to David, when he came with the Philistines against Saul to battle; but they did not help them; for the lords of the Philistines sent him away after consultation, saying, "He will fall away to his master Saul to the jeopardy of our heads."

Corpus: biomed
-- BEFORE --
In that study, there was a tendency towards correlation in transcript abundance between several pairs of antioxidant or DNA repair genes in non-BC individuals, but not in BC individuals.

This, in turn, leads to increased representation among BC individuals of individuals with lack of correlation between CEBPG and each of the affected antioxidant and/or DNA repair genes.

The 'pregnancy rate' in mice is defined as successful pregnancies per detected vaginal plug, a phenotype associated with early pregnancy failure, which in turn possibly could have an inflammatory cause.
-- AFTER --
In that study, there was a tendency towards correlation in transcript abundance between several pairs of antioxidant or DNA repair genes in non-BECAUSE individuals, but not in BECAUSE individuals.

This, in turn, leads to increased representation among BECAUSE individuals of individuals with lack of correlation between CEBPG and each of the affected antioxidant and/or DNA repair genes.

The 'pregnancy rate' in mice is defined as successful pregnancies per detected vaginal plug, a phenotype associated with early pregnancy failure, which in turn possibly could have an inflammatory because.

Corpus: europarl

-- BEFORE --
The next item is the oral question to the Commission (B7-0240/2009) by Silvia-Adriana Ţicău, Brian Simpson, János Áder, Hannes Swoboda, Eva Lichtenberger, Michael Cramer, Saïd El Khadraoui, Mathieu Grosch, Iuliu Winkler, Victor Boştinaru, Ioan Mircea Paşcu, Marian-Jean Marinescu, Ivailo Kalfin, Norica Nicolai, Dirk Sterckx, Csaba Sándor Tabajdi, Michael Theurer, Ismail Ertug, Inés Ayala Sender, Jiří Havel, Edit Herczog, Stanimir Ilchev, Iliana Malinova Iotova, Jelko Kacin, Evgeni Kirilov, Ádám Kósa, Ioan Enciu, Eduard Kukan, Gesine Meissner, Alajos Mészáros, Nadezhda Neynsky, Katarína Neveďalová, Daciana Octavia Sârbu, Vilja Savisaar, Olga Sehnalová, Catherine Stihler, Peter van Dalen, Louis Grech, Corina Creţu, George Sabin Cutaş, Vasilica Viorica Dăncilă, Cătălin Sorin Ivan, Tanja Fajon, Kinga Göncz, Antonyia Parvanova, Adina-Ioana Vălean and Rovana Plumb, on the European Strategy for the Danube Region.
-- AFTER --
The next item is the oral question to the Commission (B7-0240/2009) by Silvia-Adriana Ţicăyou, Brian Simpson, János Áder, Hannes Swoboda, Eva Lichtenberger, Michael Cramer, Saïd El Khadraoui, Mathieu Grosch, Iuliu Winkler, Victor Boştinaru, Ioan Mircea Paşcu, Marian-Jean Marinescu, Ivailo Kalfin, Norica Nicolai, Dirk Sterckx, Csaba Sándor Tabajdi, Michael Theurer, Ismail Ertug, Inés Ayala Sender, Jiří Havel, Edit Herczog, Stanimir Ilchev, Iliana Malinova Iotova, Jelko Kacin, Evgeni Kirilov, Ádám Kósa, Ioan Enciu, Eduard Kukan, Gesine Meissner, Alajos Mészáros, Nadezhda Neynsky, Katarína Neveďalová, Daciana Octavia Sârbu, Vilja Savisaar, Olga Sehnalová, Catherine Stihler, Peter van Dalen, Louis Grech, Corina Creţyou, George Sabin Cutaş, Vasilica Viorica Dăncilă, Cătălin Sorin Ivan, Tanja Fajon, Kinga Göncz, Antonyia Parvanova, Adina-Ioana Vălean and Rovana Plumb, on the European Strategy for the Danube Region.

==========
DataFrame: test_multi_df
==========

Corpus: bible
-- BEFORE --
Yet he didn't leave himself without witness, in that he did good and gave you rains from the sky and fruitful seasons, filling our hearts with food and gladness."
When he has leveled its surface, doesn't he plant the dill, and scatter the cumin seed, and put in the wheat in rows, the barley in the appointed place, and the spelt in its place?
Don't count your handmaid for a wicked woman; for I have been speaking out of the abundance of my complaint and my provocation."
-- AFTER --
Yet he did not leave himself without witness, in that he did good and gave you rains from the sky and fruitful seasons, filling our hearts with food and gladness."
When he has leveled its surface, does not he plant the dill, and scatter

the cumin seed, and put in the wheat in rows, the barley in the appointed place, and the spelt in its place?

Do not count your handmaid for a wicked woman; for I have been speaking out of the abundance of my complaint and my provocation."

[26]:
```
# check for null values

dataframes = [train_single_df, train_multi_df, trial_val_single_df,␣
 ↪trial_val_multi_df, test_single_df, test_multi_df]
for df in dataframes:
  print(df['sentence_no_contractions'].isnull().values.any())
```

```
False
False
False
False
False
False
```

[27]:
```
dataframes = {
    "train_single_df": train_single_df,
    "train_multi_df": train_multi_df,
    "trial_val_single_df": trial_val_single_df,
    "trial_val_multi_df": trial_val_multi_df,
    "test_single_df": test_single_df,
    "test_multi_df": test_multi_df
}

total_true_counts = 0
for df_name, df in dataframes.items():
    true_count = df['contraction_expanded'].sum()
    print(f"{df_name}: {true_count} True values in 'contraction_expanded'")
    total_true_counts += true_count

print(f"\nTotal True values across all dataframes: {total_true_counts}")
```

```
train_single_df: 254 True values in 'contraction_expanded'
train_multi_df: 54 True values in 'contraction_expanded'
trial_val_single_df: 16 True values in 'contraction_expanded'
trial_val_multi_df: 0 True values in 'contraction_expanded'
test_single_df: 31 True values in 'contraction_expanded'
test_multi_df: 7 True values in 'contraction_expanded'

Total True values across all dataframes: 362
```

[28]:
```
# verify column headers
```

```
dataframes = [train_single_df, train_multi_df, trial_val_single_df,␣
  ↪trial_val_multi_df, test_single_df, test_multi_df]
for df in dataframes:
  print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7662 entries, 0 to 7661
Data columns (total 7 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   id                        7662 non-null   object
 1   corpus                    7662 non-null   object
 2   sentence                  7662 non-null   object
 3   token                     7655 non-null   object
 4   complexity                7662 non-null   float64
 5   sentence_no_contractions  7662 non-null   object
 6   contraction_expanded      7662 non-null   bool
dtypes: bool(1), float64(1), object(5)
memory usage: 366.8+ KB
None
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1517 entries, 0 to 1516
Data columns (total 7 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   id                        1517 non-null   object
 1   corpus                    1517 non-null   object
 2   sentence                  1517 non-null   object
 3   token                     1517 non-null   object
 4   complexity                1517 non-null   float64
 5   sentence_no_contractions  1517 non-null   object
 6   contraction_expanded      1517 non-null   bool
dtypes: bool(1), float64(1), object(5)
memory usage: 72.7+ KB
None
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 421 entries, 0 to 420
Data columns (total 7 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   id                        421 non-null    object
 1   corpus                    421 non-null    object
 2   sentence                  421 non-null    object
 3   token                     421 non-null    object
 4   complexity                421 non-null    float64
 5   sentence_no_contractions  421 non-null    object
 6   contraction_expanded      421 non-null    bool
dtypes: bool(1), float64(1), object(5)
```

```
memory usage: 20.3+ KB
None
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 99 entries, 0 to 98
Data columns (total 7 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   id                        99 non-null     object
 1   corpus                    99 non-null     object
 2   sentence                  99 non-null     object
 3   token                     99 non-null     object
 4   complexity                99 non-null     float64
 5   sentence_no_contractions  99 non-null     object
 6   contraction_expanded      99 non-null     bool
dtypes: bool(1), float64(1), object(5)
memory usage: 4.9+ KB
None
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 917 entries, 0 to 916
Data columns (total 7 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   id                        917 non-null    object
 1   corpus                    917 non-null    object
 2   sentence                  917 non-null    object
 3   token                     917 non-null    object
 4   complexity                917 non-null    float64
 5   sentence_no_contractions  917 non-null    object
 6   contraction_expanded      917 non-null    bool
dtypes: bool(1), float64(1), object(5)
memory usage: 44.0+ KB
None
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 184 entries, 0 to 183
Data columns (total 7 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   id                        184 non-null    object
 1   corpus                    184 non-null    object
 2   sentence                  184 non-null    object
 3   token                     184 non-null    object
 4   complexity                184 non-null    float64
 5   sentence_no_contractions  184 non-null    object
 6   contraction_expanded      184 non-null    bool
dtypes: bool(1), float64(1), object(5)
memory usage: 8.9+ KB
None
```

[29]:
```
# inspect each df

dataframes = [train_single_df, train_multi_df, trial_val_single_df,␣
 ↪trial_val_multi_df, test_single_df, test_multi_df]
for df in dataframes:
  print(df.head())
```

```
                              id corpus
sentence     token  complexity
sentence_no_contractions  contraction_expanded
0  3ZLW647WALVGE8EBR5OEGUBPU4P32A  bible  Behold, there came up out of the river
seven c…     river    0.000000  Behold, there came up out of the river seven
c…                  False
1  34ROBODSP1ZBN3DVY8J8XSIY551E5C  bible  I am a fellow bondservant with you and
with yo…  brothers    0.000000  I am a fellow bondservant with you and with
yo…                 False
2  3S1WOPCJFGTJU2SGNAN2Y213N6WJE3  bible  The man, the lord of the land, said to
us, 'By…  brothers    0.050000  The man, the lord of the land, said to us,
'By…                 False
3  3BFNCI9LYKQNO9BHXHH9CLSX5KP738  bible  Shimei had sixteen sons and six
daughters; but…  brothers    0.150000  Shimei had sixteen sons and six
daughters; but…                  True
4  3G5RUKN2EC3YIWSKUXZ8ZVH95R49N2  bible              "He has put my brothers
far from me.  brothers    0.263889              "He has put my brothers far
from me.                False
                              id corpus
sentence            token  complexity
sentence_no_contractions  contraction_expanded
0  3S37Y8CWI8ON8KVM53U4E6JKCDC4WE  bible  but the seventh day is a Sabbath to
Yahweh you…      seventh day    0.027778  but the seventh day is a Sabbath to
Yahweh you…                 False
1  3WGCNLZJKF877FYC1Q6COKNWTDWD11  bible  But let each man test his own work,
and then h…         own work    0.050000  But let each man test his own work,
and then h…                 False
2  3UOMW19E6D6WQ5TH2HDD74IVKTP5CB  bible  To him who by understanding made the
heavens; …  loving kindness    0.050000  To him who by understanding made the
heavens; …                  False
3  36JW4WBRO6KF9AXMUL4N476OMF8FHD  bible  Remember to me, my God, this also, and
spare m…  loving kindness    0.050000  Remember to me, my God, this also, and
spare m…                 False
4  3HRWUH63QU2FH9Q8R7MRNFC7JX2N5A  bible  Because your loving kindness is better
than li…  loving kindness    0.075000  Because your loving kindness is better
than li…                 False
                              id corpus
sentence token  complexity                          sentence_no_contractions
contraction_expanded
0  3QI9WAYOGQB8GQIR4MDIEFOD2RLS67  bible  They will not hurt nor destroy in all
my holy …   sea    0.000000  They will not hurt nor destroy in all my holy …
```

```
False
1  3T8DUCXYON6WD9X4RTLK8UN1U929TF  bible  that sends ambassadors by the sea,
even in ves…   sea    0.102941  that sends ambassadors by the sea, even in
ves…                  False
2  3I7KR83SNADXAQ7HXK7S7305BYB9KD  bible  and they entered into the boat, and
were going…   sea    0.109375  and they entered into the boat, and were
going…                  False
3  3BO3NEOQMOHK9ERYPNOGQIWCPC4IAQ  bible  Joseph laid up grain as the sand of
the sea, v…   sea    0.160714  Joseph laid up grain as the sand of the sea,
v…                 False
4  3Y3CZJSZ9KTOW7IOKE38WZHHKSW5RH  bible  There will be a highway for the
remnant that i…  land    0.000000  There will be a highway for the remnant
that i…                 False
                                id corpus
sentence          token  complexity
sentence_no_contractions  contraction_expanded
0  31HLTCK4BLVQ5BO1AUR91TX9V9IVGH  bible  The name of one son was Gershom, for
Moses sai…   foreign land    0.000000  The name of one son was Gershom, for
Moses sai…                 False
1  389A2A304OIXVY7G5B71Q9M43LEOCL  bible  unleavened bread, unleavened cakes
mixed with …    wheat flour    0.157895  unleavened bread, unleavened cakes
mixed with …                 False
2  31N9JPQXIPIRX2A3S9NOCCFXO6TNHR  bible  However the high places were not taken
away; t…  burnt incense    0.200000  However the high places were not taken
away; t…                 False
3  3JVP4ZJHDPSO81TGXL3N1CKZGQYOIN  bible  and he burnt incense of sweet spices
on it, as…  burnt incense    0.250000  and he burnt incense of sweet spices on
it, as…                 False
4  3JAOYN9IHL25ZQAUV5EJZ4GHOKL33L  bible  The same day the king made the middle
of the c…   bronze altar    0.214286  The same day the king made the middle of
the c…                 False
                                id corpus
sentence     token  complexity
sentence_no_contractions  contraction_expanded
0  3K8CQCU3KE19US5SN89ODFPK3SANWR  bible  But he, beckoning to them with his
hand to be …      hand    0.000000  But he, beckoning to them with his hand to
be …                 False
1  3Q2T3FDOON86LCI41NJYV3PNOBW3MV  bible  If I forget you, Jerusalem, let my
right hand …      hand    0.197368  If I forget you, Jerusalem, let my right
hand …                 False
2  3ULIZOH1VA5C32JJMKOTQ8Z4GUS51B  bible  the ten sons of Haman the son of
Hammedatha, t…      hand    0.200000  the ten sons of Haman the son of
Hammedatha, t…                  True
3  3BFFODJK8XCEIOT3OZLBPPSRMZQTSD  bible  Let your hand be lifted up above your
adversar…      hand    0.267857  Let your hand be lifted up above your
adversar…                 False
4  3QREJ3J433XSBS8QMHAICCROBQ1LKR  bible  Abimelech chased him, and he fled
before him, …  entrance    0.000000  Abimelech chased him, and he fled before
```

```
him, …                  False
                                id corpus
sentence           token  complexity
sentence_no_contractions  contraction_expanded
0  3UXQ63NLAAMRIP4WG4XPD98AOYOBLX  bible  for he had an only daughter, about
twelve year…   only daughter    0.025000  for he had an only daughter, about
twelve year…                  False
1  3FJ2RVH25Z62TA3R8E1O77EBUYU92W  bible  All these were cities fortified with
high wall…      high walls    0.100000  All these were cities fortified with
high wall…                  False
2  3YO4AH2FPDK1PZHZAT8WAEBL7OEQOF  bible  In the morning, 'It will be foul
weather today…   weather today    0.125000  In the morning, 'It will be foul
weather today…                  False
3  3X52SWXEOX5Q3O81YIOMX4V84QTCWZ  bible  Her young children also were dashed in
pieces …  young children    0.160714  Her young children also were dashed in
pieces …                  False
4  32K26U12DNONTREA84Q1V8UCIH2VD7  bible  All king Solomon's drinking vessels
were of go…       pure gold    0.178571  All king Solomon's drinking vessels
were of go…                  False
```

[30]:
```python
tokenizer = RegexpTokenizer(r'\w+')

def analyze_sentence_spans_by_corpus_and_quartile_no_contracts(dfs_dict):
    """
    Analyze sentence spans (length metrics) grouped by corpus and complexity␣
 ↪quartile
    for multiple dataframes, but this time using the 'sentence_no_contractions'␣
 ↪column
    instead of the original 'sentence'.
    """
    results = []

    for df_name, df in dfs_dict.items():
        print(f"Processing {df_name} on 'sentence_no_contractions'...")
        df = df.copy()

        q1 = df['complexity'].quantile(0.25)
        q2 = df['complexity'].quantile(0.50)
        q3 = df['complexity'].quantile(0.75)

        def get_quartile(x):
            if x <= q1:
                return 'Q1'
            elif x <= q2:
                return 'Q2'
            elif x <= q3:
                return 'Q3'
```

```
            else:
                return 'Q4'

        df['quartile'] = df['complexity'].apply(get_quartile)

        def compute_span_metrics_no_contracts(sentence):
            if pd.isna(sentence):
                return pd.Series({'word_count': 0, 'char_count': 0,␣
↪'avg_word_len': 0})

            words = tokenizer.tokenize(sentence)
            word_count = len(words)
            char_count = len(sentence)
            avg_word_len = np.mean([len(w) for w in words]) if word_count > 0␣
↪else 0

            return pd.Series({
                'word_count': word_count,
                'char_count': char_count,
                'avg_word_len': avg_word_len
            })

        span_metrics_nc = df['sentence_no_contractions'].
↪apply(compute_span_metrics_no_contracts)
        df = pd.concat([df, span_metrics_nc], axis=1)

        corpus_col = 'corpus'
        for corpus_name, corpus_df in df.groupby(corpus_col):
            for quartile, quartile_df in corpus_df.groupby('quartile'):
                complexity_range = f"{quartile_df['complexity'].min():.
↪3f}-{quartile_df['complexity'].max():.3f}"
                stats = {
                    'Dataframe': df_name,
                    'Corpus': corpus_name,
                    'Quartile': quartile,
                    'Complexity Range': complexity_range,
                    'Count': len(quartile_df),
                    'Avg Words': quartile_df['word_count'].mean(),
                    'Median Words': quartile_df['word_count'].median(),
                    'Min Words': quartile_df['word_count'].min(),
                    'Max Words': quartile_df['word_count'].max(),
                    'Std Words': quartile_df['word_count'].std(),
                    'Avg Chars': quartile_df['char_count'].mean(),
                    'Avg Word Len': quartile_df['avg_word_len'].mean()
                }
                results.append(stats)
```

```
    results_df = pd.DataFrame(results)
    results_df = results_df.sort_values(['Dataframe', 'Corpus', 'Quartile'])
    return results_df


dfs = {
    'train_single_df': train_single_df,
    'train_multi_df': train_multi_df,
    'trial_val_single_df': trial_val_single_df,
    'trial_val_multi_df': trial_val_multi_df,
    'test_single_df': test_single_df,
    'test_multi_df': test_multi_df
}

span_analysis_nc =␣
 ↪analyze_sentence_spans_by_corpus_and_quartile_no_contracts(dfs)

pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)
pd.set_option('display.width', 1000)
display(span_analysis_nc)

results_path_nc = os.path.join(dir_results,␣
 ↪'sentence_span_analysis_no_contractions.csv')
span_analysis_nc.to_csv(results_path_nc, index=False)
print(f"Analysis (NO CONTRACTIONS) saved to: {results_path_nc}")
```

```
Processing train_single_df on 'sentence_no_contractions'…
Processing train_multi_df on 'sentence_no_contractions'…
Processing trial_val_single_df on 'sentence_no_contractions'…
Processing trial_val_multi_df on 'sentence_no_contractions'…
Processing test_single_df on 'sentence_no_contractions'…
Processing test_multi_df on 'sentence_no_contractions'…
```

	Dataframe	Corpus	Quartile	Complexity Range	Count	Avg Words	Median Words	Min Words	Max Words	Std Words	Avg Chars	Avg Word Len
60	test_multi_df	bible	Q1	0.025-0.317	26	23.076923	22.0	4.0	48.0	11.831900	118.730769	4.131249
61	test_multi_df	bible	Q2	0.325-0.417	11	20.545455	17.0	7.0	47.0	12.917923	109.636364	4.213539
62	test_multi_df	bible	Q3	0.432-0.528	18	21.055556	21.5	4.0	43.0	10.843660	113.166667	4.498610
63	test_multi_df	bible	Q4	0.542-0.694	11	22.363636	20.0	7.0	51.0	11.935432	126.181818	4.605062
64	test_multi_df	biomed	Q1	0.000-0.312	11	29.818182	29.0	17.0	47.0	8.388304	195.727273	5.491145
65	test_multi_df	biomed	Q2	0.324-0.417	11	27.090909	24.0	9.0	47.0	11.449494	171.818182	5.436237

```
66        test_multi_df    biomed       Q3      0.456-0.528     10   26.900000 ␣
↪      26.5        10.0        49.0  10.712921  177.500000       5.497409
67        test_multi_df    biomed       Q4      0.562-0.800     21   32.285714 ␣
↪      34.0        14.0        56.0  13.598319  209.285714       5.460101
68        test_multi_df  europarl       Q1      0.214-0.303     10   24.700000 ␣
↪      24.5         7.0        56.0  14.189589  146.900000       5.049688
69        test_multi_df  europarl       Q2      0.321-0.429     24   27.833333 ␣
↪      27.0         9.0        73.0  15.352855  172.291667       5.269610
70        test_multi_df  europarl       Q3      0.432-0.516     18   32.944444 ␣
↪      32.0         6.0        68.0  19.129504  209.888889       5.512245
71        test_multi_df  europarl       Q4      0.531-0.562     13   39.000000 ␣
↪      36.0         6.0        95.0  29.631065  237.076923       5.100616
48       test_single_df     bible       Q1      0.000-0.214     79   22.822785 ␣
↪      22.0         7.0        49.0  10.585137  116.924051       4.040893
49       test_single_df     bible       Q2      0.217-0.276     68   24.176471 ␣
↪      21.0         2.0        77.0  14.393138  126.088235       4.172273
50       test_single_df     bible       Q3      0.278-0.353     67   22.388060 ␣
↪      20.0         4.0        63.0  11.306950  119.776119       4.256042
51       test_single_df     bible       Q4      0.359-0.732     69   20.579710 ␣
↪      19.0         1.0        55.0  11.264736  110.637681       4.341070
52       test_single_df    biomed       Q1      0.000-0.214     75   27.080000 ␣
↪      25.0        10.0        84.0  12.025603  172.986667       5.277318
53       test_single_df    biomed       Q2      0.217-0.275     58   30.275862 ␣
↪      26.0        10.0        83.0  15.856587  198.293103       5.446788
54       test_single_df    biomed       Q3      0.278-0.357     66   29.833333 ␣
↪      29.0        13.0        85.0  11.754650  191.863636       5.334048
55       test_single_df    biomed       Q4      0.359-0.778     90   31.144444 ␣
↪      30.0        14.0        83.0  12.089146  203.077778       5.393791
56       test_single_df  europarl       Q1      0.000-0.214     83   25.337349 ␣
↪      21.0         3.0        82.0  16.032191  151.891566       5.044222
57       test_single_df  europarl       Q2      0.217-0.276     98   32.326531 ␣
↪      30.0         1.0        97.0  18.707061  195.653061       5.062296
58       test_single_df  europarl       Q3      0.278-0.357     96   33.000000 ␣
↪      30.0         3.0       141.0  21.404377  201.760417       5.124551
59       test_single_df  europarl       Q4      0.361-0.583     68   33.235294 ␣
↪      29.0         1.0       130.0  20.440023  206.573529       5.164576
12       train_multi_df     bible       Q1      0.028-0.300    163   23.570552 ␣
↪      22.0         3.0        67.0  12.429043  124.871166       4.237932
13       train_multi_df     bible       Q2      0.304-0.409    132   24.053030 ␣
↪      22.0         6.0        65.0  11.738444  129.659091       4.305703
14       train_multi_df     bible       Q3      0.411-0.529    131   23.778626 ␣
↪      23.0         4.0        50.0  11.179163  127.564885       4.331458
15       train_multi_df     bible       Q4      0.533-0.778     79   25.481013 ␣
↪      24.0         3.0        81.0  13.490605  139.405063       4.491816
16       train_multi_df    biomed       Q1      0.028-0.303     87   29.091954 ␣
↪      28.0         9.0        77.0  11.882792  185.977011       5.277384
```

```
17       train_multi_df    biomed       Q2      0.304-0.408     74  30.756757 ␣
 ↪      28.0       11.0       85.0  13.511853  196.067568      5.367302
18       train_multi_df    biomed       Q3      0.411-0.529    111  29.783784 ␣
 ↪      29.0        8.0       61.0  10.912383  193.873874      5.430754
19       train_multi_df    biomed       Q4      0.531-0.975    242  29.607438 ␣
 ↪      28.0       10.0       75.0  12.029995  195.107438      5.535387
20       train_multi_df  europarl       Q1      0.118-0.303    132  29.363636 ␣
 ↪      27.0        3.0      101.0  17.874146  176.583333      5.003685
21       train_multi_df  europarl       Q2      0.304-0.409    171  31.666667 ␣
 ↪      28.0        3.0      108.0  19.112977  195.198830      5.176456
22       train_multi_df  europarl       Q3      0.411-0.529    138  33.398551 ␣
 ↪      30.0        7.0      101.0  18.992715  208.304348      5.286607
23       train_multi_df  europarl       Q4      0.533-0.750     57  34.596491 ␣
 ↪      31.0        6.0       96.0  20.318763  218.350877      5.345891
0       train_single_df     bible       Q1      0.000-0.212    701  23.269615 ␣
 ↪      22.0        4.0       61.0  11.764113  121.714693      4.135685
1       train_single_df     bible       Q2      0.212-0.279    640  23.750000 ␣
 ↪      22.0        3.0       60.0  11.579622  124.671875      4.153925
2       train_single_df     bible       Q3      0.281-0.375    624  23.825321 ␣
 ↪      22.0        3.0       70.0  11.963291  126.338141      4.213931
3       train_single_df     bible       Q4      0.380-0.861    609  23.586207 ␣
 ↪      21.0        3.0       69.0  12.460182  126.602627      4.298065
4       train_single_df    biomed       Q1      0.000-0.212    586  28.534130 ␣
 ↪      27.0        2.0       85.0  12.115387  182.076792      5.322266
5       train_single_df    biomed       Q2      0.212-0.279    583  30.442539 ␣
 ↪      29.0        7.0       92.0  11.863182  193.921098      5.289166
6       train_single_df    biomed       Q3      0.281-0.375    659  29.860395 ␣
 ↪      28.0        4.0       77.0  11.591263  191.098634      5.329940
7       train_single_df    biomed       Q4      0.381-0.861    748  29.181818 ␣
 ↪      28.0        3.0       85.0  12.249267  186.978610      5.299963
8       train_single_df  europarl       Q1      0.025-0.212    641  26.761310 ␣
 ↪      24.0        2.0      107.0  15.230853  159.190328      4.942926
9       train_single_df  europarl       Q2      0.212-0.279    714  30.420168 ␣
 ↪      27.0        1.0      129.0  18.383783  183.105042      4.995897
10      train_single_df  europarl       Q3      0.281-0.375    701  30.523538 ␣
 ↪      28.0        1.0      122.0  18.163026  185.843081      5.114626
11      train_single_df  europarl       Q4      0.381-0.775    456  33.543860 ␣
 ↪      31.0        2.0      235.0  21.708515  203.664474      5.054387
36   trial_val_multi_df     bible       Q1      0.000-0.292     11  26.272727 ␣
 ↪      21.0       13.0       64.0  13.950562  141.363636      4.282457
37   trial_val_multi_df     bible       Q2      0.333-0.400      7  20.571429 ␣
 ↪      23.0        5.0       28.0   7.412987  110.857143      4.279406
38   trial_val_multi_df     bible       Q3      0.425-0.500      5  19.600000 ␣
 ↪      19.0        9.0       32.0   8.905055  109.200000      4.431391
39   trial_val_multi_df     bible       Q4      0.525-0.661      6  22.333333 ␣
 ↪      20.5        9.0       44.0  12.242004  117.833333      4.178525
```

```
40   trial_val_multi_df    biomed       Q1      0.083-0.303      6  26.833333
 ↪      25.0       15.0       49.0  11.771434  159.166667      4.899969
41   trial_val_multi_df    biomed       Q2      0.317-0.422      7  25.428571
 ↪      21.0       15.0       48.0  11.588171  156.000000      5.194383
42   trial_val_multi_df    biomed       Q3      0.438-0.513      6  37.833333
 ↪      39.5       26.0       44.0   6.675827  247.500000      5.438593
43   trial_val_multi_df    biomed       Q4      0.537-0.825     14  30.642857
 ↪      29.5       17.0       43.0   9.849695  211.428571      5.730623
44   trial_val_multi_df  europarl       Q1      0.176-0.306      8  30.000000
 ↪      25.5        4.0       64.0  20.361027  186.750000      5.306837
45   trial_val_multi_df  europarl       Q2      0.312-0.412     11  47.909091
 ↪      46.0       24.0       78.0  18.651834  296.909091      5.058375
46   trial_val_multi_df  europarl       Q3      0.432-0.500     13  26.307692
 ↪      26.0        5.0       66.0  18.167666  166.153846      5.263847
47   trial_val_multi_df  europarl       Q4      0.515-0.714      5  26.400000
 ↪      15.0        6.0       66.0  24.316661  164.600000      4.998182
24  trial_val_single_df     bible       Q1      0.000-0.214     52  26.769231
 ↪      26.0        5.0       74.0  15.589860  137.423077      4.074456
25  trial_val_single_df     bible       Q2      0.217-0.266     38  24.868421
 ↪      23.0        7.0       50.0  10.768249  131.342105      4.200230
26  trial_val_single_df     bible       Q3      0.268-0.355     26  22.884615
 ↪      20.5        5.0       44.0   9.961233  121.423077      4.316593
27  trial_val_single_df     bible       Q4      0.361-0.633     27  25.666667
 ↪      23.0        6.0       49.0  12.554497  137.592593      4.213842
28  trial_val_single_df    biomed       Q1      0.028-0.214     21  25.571429
 ↪      21.0       13.0       65.0  11.543706  164.380952      5.317614
29  trial_val_single_df    biomed       Q2      0.217-0.267     28  30.571429
 ↪      27.5       11.0       57.0  12.099674  198.142857      5.315287
30  trial_val_single_df    biomed       Q3      0.268-0.359     38  32.105263
 ↪      29.0       11.0       61.0  12.710476  206.947368      5.364934
31  trial_val_single_df    biomed       Q4      0.364-0.875     48  25.145833
 ↪      25.5        6.0       56.0  11.721937  164.020833      5.445661
32  trial_val_single_df  europarl       Q1      0.050-0.214     33  31.969697
 ↪      28.0        5.0       81.0  20.356947  185.969697      4.799024
33  trial_val_single_df  europarl       Q2      0.217-0.267     41  28.487805
 ↪      28.0        4.0       71.0  15.424205  172.902439      4.997384
34  trial_val_single_df  europarl       Q3      0.268-0.359     39  30.282051
 ↪      28.0        3.0       99.0  20.040681  184.358974      5.086945
35  trial_val_single_df  europarl       Q4      0.367-0.605     30  35.700000
 ↪      30.5        5.0       77.0  20.142852  215.400000      4.910759

Analysis (NO CONTRACTIONS) saved to: /content/drive/MyDrive/266-
final/results/sentence_span_analysis_no_contractions.csv
```

- contraction processing successfuly, confirmed with Avg Word deltas between 'sentence' and 'sentence_no_contractions'

0.4 Enrich Datset with PoS Tags, Dependency Parsing, and Morphological Complexity

[31]:
```
# !pip install -q spacy
# !python -m spacy download en_core_web_trf
!python -m spacy download en_core_web_lg
```

```
Collecting en-core-web-lg==3.8.0
  Downloading https://github.com/explosion/spacy-
models/releases/download/en_core_web_lg-3.8.0/en_core_web_lg-3.8.0-py3-none-
any.whl (400.7 MB)
                           400.7/400.7
MB 2.6 MB/s eta 0:00:00
Installing collected packages: en-core-web-lg
Successfully installed en-core-web-lg-3.8.0
 Download and installation successful
You can now load the package via spacy.load('en_core_web_lg')
 Restart to reload dependencies
If you are in a Jupyter or Colab notebook, you may need to restart Python in
order to load all the package's dependencies. You can do this by selecting the
'Restart kernel' or 'Restart runtime' option.
```

[32]:
```
nlp = spacy.load("en_core_web_lg")
```

[33]:
```
text = "This is a sample sentence for testing spaCy."

doc = nlp(text)

for token in doc:
    print(f"Token: {token.text}, POS: {token.pos_}, Dependency: {token.dep_}")
```

```
Token: This, POS: PRON, Dependency: nsubj
Token: is, POS: AUX, Dependency: ROOT
Token: a, POS: DET, Dependency: det
Token: sample, POS: NOUN, Dependency: compound
Token: sentence, POS: NOUN, Dependency: attr
Token: for, POS: ADP, Dependency: prep
Token: testing, POS: VERB, Dependency: pcomp
Token: spaCy, POS: PROPN, Dependency: dobj
Token: ., POS: PUNCT, Dependency: punct
```

[34]:
```
def enrich_with_spacy(df, text_col='sentence_no_contractions'):
    """
    Processes the 'text_col' with spaCy and appends:
      pos_sequence, dep_sequence, morph_sequence,
      and morph_complexity (float) per row.
    """
    df = df.copy()
```

```
    pos_tags = []
    dep_tags = []
    morph_tags = []
    morph_complexities = []

    for text in df[text_col]:
        if pd.isna(text) or not text.strip():
            pos_tags.append([])
            dep_tags.append([])
            morph_tags.append([])
            morph_complexities.append(0.0)
            continue

        doc = nlp(text)

        pos_seq = [token.pos_ for token in doc]
        dep_seq = [token.dep_ for token in doc]
        morph_seq = [token.morph for token in doc]

        total_features = 0
        for token in doc:
            features_dict = token.morph.to_dict()
            total_features += len(features_dict)

        avg_morph = total_features / len(doc)

        pos_tags.append(pos_seq)
        dep_tags.append(dep_seq)
        morph_tags.append(morph_seq)
        morph_complexities.append(avg_morph)

    df['pos_sequence'] = pos_tags
    df['dep_sequence'] = dep_tags
    df['morph_sequence'] = morph_tags
    df['morph_complexity'] = morph_complexities

    return df
```

[ ]:
```
dataframes_info = [
    ("train_single_df", train_single_df),
    ("train_multi_df", train_multi_df),
    ("trial_val_single_df", trial_val_single_df),
    ("trial_val_multi_df", trial_val_multi_df),
    ("test_single_df", test_single_df),
    ("test_multi_df", test_multi_df),
]
```

```
for df_name, df in dataframes_info:
    print(f"Enriching {df_name} with spaCy features...")
    enriched_df = enrich_with_spacy(df, text_col='sentence_no_contractions')
    globals()[df_name] = enriched_df
    print(f"Done! Now '{df_name}' has columns: pos_sequence, dep_sequence,␣
 ↪morph_sequence, morph_complexity.\n")
```

Enriching train_single_df with spaCy features…

[ ]:
```
for df_name, df in dataframes_info:
    print(f"\n{'='*50}")
    print(f"DataFrame: {df_name}")
    print(f"{'='*50}\n")
    sample_df = globals()[df_name].sample(3, random_state=42)
    display(sample_df[['sentence_no_contractions', 'pos_sequence',␣
 ↪'dep_sequence', 'morph_sequence', 'morph_complexity']])
```

[ ]:
```
# verify column headers

dataframes = [train_single_df, train_multi_df, trial_val_single_df,␣
 ↪trial_val_multi_df, test_single_df, test_multi_df]
for df in dataframes:
  print(df.info())
```

0.5 Create Binarized Outcome Variable, based on train_single_df median and train_multi_df median, applied to trial-val and test

[ ]:
```
train_single_median = train_single_df['complexity'].median()

def binarize_complexity(value, threshold):
    """
    If value <= threshold, return 0, else return 1.
    """
    if value <= threshold:
        return 0
    else:
        return 1

train_single_df['binary_complexity'] = train_single_df['complexity'].apply(
    lambda x: binarize_complexity(x, train_single_median)
)
trial_val_single_df['binary_complexity'] = trial_val_single_df['complexity'].
 ↪apply(
    lambda x: binarize_complexity(x, train_single_median)
)
test_single_df['binary_complexity'] = test_single_df['complexity'].apply(
```

```
    lambda x: binarize_complexity(x, train_single_median)
)

train_multi_median = train_multi_df['complexity'].median()

train_multi_df['binary_complexity'] = train_multi_df['complexity'].apply(
    lambda x: binarize_complexity(x, train_multi_median)
)
trial_val_multi_df['binary_complexity'] = trial_val_multi_df['complexity'].
 ↪apply(
    lambda x: binarize_complexity(x, train_multi_median)
)
test_multi_df['binary_complexity'] = test_multi_df['complexity'].apply(
    lambda x: binarize_complexity(x, train_multi_median)
)

print(f"Median complexity (single): {train_single_median}")
print(f"Median complexity (multi): {train_multi_median}")

print("\nSample rows from train_single_df:")
print(train_single_df[['id', 'complexity', 'binary_complexity']].head())

print("\nSample rows from train_multi_df:")
print(train_multi_df[['id', 'complexity', 'binary_complexity']].head())
```

[ ]:
```
# verify column headers

dataframes = [train_single_df, train_multi_df, trial_val_single_df,␣
 ↪trial_val_multi_df, test_single_df, test_multi_df]
for df in dataframes:
  print(df.info())
```

[ ]:
```
# inspect each df

dataframes = [train_single_df, train_multi_df, trial_val_single_df,␣
 ↪trial_val_multi_df, test_single_df, test_multi_df]
for df in dataframes:
  print(df.head())
```

[ ]:
```
dataframes = {
    "train_single_df": train_single_df,
    "train_multi_df": train_multi_df,
    "trial_val_single_df": trial_val_single_df,
    "trial_val_multi_df": trial_val_multi_df,
    "test_single_df": test_single_df,
    "test_multi_df": test_multi_df
}
```

```
fig, axes = plt.subplots(2, 3, figsize=(18, 12))

for i, (df_name, df) in enumerate(dataframes.items()):
  row = i // 3
  col = i % 3
  ax = axes[row, col]
  sns.histplot(df['binary_complexity'], kde=True, ax=ax)
  ax.set_title(f'Distribution of binary_complexity for {df_name}')
  ax.set_xlabel('binary_complexity')

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

[ ]:
```
train_single_75th = train_single_df['complexity'].quantile(0.75)
train_multi_75th = train_multi_df['complexity'].quantile(0.75)

print("75th percentile (single-track):", train_single_75th)
print("75th percentile (multi-track):", train_multi_75th)

def binarize_complexity_75th(value, threshold):
    """
    Returns 0 if 'value' <= threshold, else 1.
    """
    if value <= threshold:
        return 0
    else:
        return 1

train_single_df['binary_complexity_75th_split'] = train_single_df['complexity'].
 ↪apply(
    lambda x: binarize_complexity_75th(x, train_single_75th)
)
trial_val_single_df['binary_complexity_75th_split'] =␣
 ↪trial_val_single_df['complexity'].apply(
    lambda x: binarize_complexity_75th(x, train_single_75th)
)
test_single_df['binary_complexity_75th_split'] = test_single_df['complexity'].
 ↪apply(
    lambda x: binarize_complexity_75th(x, train_single_75th)
)

train_multi_df['binary_complexity_75th_split'] = train_multi_df['complexity'].
 ↪apply(
    lambda x: binarize_complexity_75th(x, train_multi_75th)
)
```

```
trial_val_multi_df['binary_complexity_75th_split'] =␣
 ↪trial_val_multi_df['complexity'].apply(
    lambda x: binarize_complexity_75th(x, train_multi_75th)
)
test_multi_df['binary_complexity_75th_split'] = test_multi_df['complexity'].
 ↪apply(
    lambda x: binarize_complexity_75th(x, train_multi_75th)
)

print("\nDistribution of 'binary_complexity_75th_split' in train_single_df:")
print(train_single_df['binary_complexity_75th_split'].value_counts())

print("\nDistribution of 'binary_complexity_75th_split' in train_multi_df:")
print(train_multi_df['binary_complexity_75th_split'].value_counts())
```

[ ]:
```
dataframes = {
    "train_single_df": train_single_df,
    "train_multi_df": train_multi_df,
    "trial_val_single_df": trial_val_single_df,
    "trial_val_multi_df": trial_val_multi_df,
    "test_single_df": test_single_df,
    "test_multi_df": test_multi_df
}

fig, axes = plt.subplots(2, 3, figsize=(18, 12))

for i, (df_name, df) in enumerate(dataframes.items()):
  row = i // 3
  col = i % 3
  ax = axes[row, col]
  sns.histplot(df['binary_complexity_75th_split'], kde=True, ax=ax)
  ax.set_title(f'Distribution of binary_complexity_75th_split for {df_name}')
  ax.set_xlabel('binary_complexity_75th_split')

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

[ ]:
```
!ls -R /content/drive/MyDrive/266-final/data/266-comp-lex-master/
```

[ ]:
```
# !tree /content/drive/MyDrive/266-final/data/266-comp-lex-master/
```

[ ]:
```
import os
dataframes = {
    "train_single_df": train_single_df,
    "train_multi_df": train_multi_df,
    "trial_val_single_df": trial_val_single_df,
    "trial_val_multi_df": trial_val_multi_df,
```

```
    "test_single_df": test_single_df,
    "test_multi_df": test_multi_df
}

base_dir = "/content/drive/MyDrive/266-final/data/266-comp-lex-master/"

for df_name, df in dataframes.items():
    subdir = None
    if "train" in df_name:
      subdir = "fe-train"
    elif "trial_val" in df_name:
      subdir = "fe-trial-val"
    elif "test" in df_name:
      subdir = "fe-test-labels"

    if subdir:
      save_path = os.path.join(base_dir, subdir, f"{df_name}.csv")
      os.makedirs(os.path.dirname(save_path), exist_ok=True)
      df.to_csv(save_path, index=False)
      print(f"Saved {df_name} to {save_path}")
```

[ ]:
```
!tree /content/drive/MyDrive/266-final/data/266-comp-lex-master/
```

[ ]:
```
print(dir_data)
```

[ ]:
```
df_names = [
    "train_single_df",
    "train_multi_df",
    "trial_val_single_df",
    "trial_val_multi_df",
    "test_single_df",
    "test_multi_df"
]

loaded_dataframes = {}

for df_name in df_names:
    if "train" in df_name:
        subdir = "fe-train"
    elif "trial_val" in df_name:
        subdir = "fe-trial-val"
    elif "test" in df_name:
        subdir = "fe-test-labels"
    else:
        subdir = None

    if subdir:
```

```
        read_path = os.path.join(dir_data, subdir, f"{df_name}.csv")
        loaded_df = pd.read_csv(read_path)
        loaded_dataframes[df_name] = loaded_df
        print(f"Loaded {df_name} from {read_path}")

for df_name, df in loaded_dataframes.items():
    print(f"\n>>> {df_name} shape: {df.shape}")
    if 'binary_complexity' in df.columns:
        print(df['binary_complexity'].value_counts())
```

[ ]:
```
# verify column headers

dataframes = [train_single_df, train_multi_df, trial_val_single_df,␣
 ↪trial_val_multi_df, test_single_df, test_multi_df]
for df in dataframes:
  print(df.info())
```

[ ]:
```
# inspect each df

dataframes = [train_single_df, train_multi_df, trial_val_single_df,␣
 ↪trial_val_multi_df, test_single_df, test_multi_df]
for df in dataframes:
  print(df.head())
```

[ ]:
```
dataframes = {
    "train_single_df": train_single_df,
    "train_multi_df": train_multi_df,
    "trial_val_single_df": trial_val_single_df,
    "trial_val_multi_df": trial_val_multi_df,
    "test_single_df": test_single_df,
    "test_multi_df": test_multi_df
}

for df_name, df in dataframes.items():
    print(f"\n=== {df_name} ===")
    print(df['binary_complexity'].value_counts())
```

0.5.1 Create sentence_no_contractions + PoS, and sentence_no_contractions + morph

[ ]:
```

```

[ ]:
```

```

[ ]:
```
!tree /content/drive/MyDrive/266-final/data/266-comp-lex-master/
```

[ ]:
```
import os
dataframes = {
    "train_single_df": train_single_df,
    "train_multi_df": train_multi_df,
    "trial_val_single_df": trial_val_single_df,
    "trial_val_multi_df": trial_val_multi_df,
    "test_single_df": test_single_df,
    "test_multi_df": test_multi_df
}

base_dir = "/content/drive/MyDrive/266-final/data/266-comp-lex-master/"

for df_name, df in dataframes.items():
    subdir = None
    if "train" in df_name:
      subdir = "fe-train"
    elif "trial_val" in df_name:
      subdir = "fe-trial-val"
    elif "test" in df_name:
      subdir = "fe-test-labels"

    if subdir:
      save_path = os.path.join(base_dir, subdir, f"{df_name}.csv")
      os.makedirs(os.path.dirname(save_path), exist_ok=True)
      df.to_csv(save_path, index=False)
      print(f"Saved {df_name} to {save_path}")
```

[ ]:
```
!tree /content/drive/MyDrive/266-final/data/266-comp-lex-master/
```

- These counts match my offline calculations exactly.The binarized outcome variables have been split on on the median of the TRAIN_SINGLE and TRAIN_MULTI dataset splits ONLY, thus this median is applied to trial_val and test. The first two quartiles (up to the train median) are equal to 0 in 'binary_complexity' and the next two quartiles are equal to 1.
- Because the dataset has been excellently balanced by the Task's annotators, we're lucky that no further data processing is necessary prior to moving onto the modeling step, and ensuring protection from data leakage by (later) removing necessary columns prior to vectorization.
- Lastly, a note on the balanced nature of the data. It should be noted that (even in the continuous outome representation of 'complexity') the medians were 0.28 in train_single, and 0.27 in both trial_single and test_single—for multi, it was 0.41 in train_multi, and 0.42 in trial_multi and 0.43 in test_multi.
- We also find that after Data Engineering, our sanity checks have come out successfully. No records have been lost, shapes are consistent with our expectations, and we have enriched the dataset with SpaCy-derived features to give us flexibility in multi-channel inputs or vectorization ablations. This is a very thorough dataset, and we are now ready for modeling.

[ ]: