creativity

March 10, 2025

[ ]:
```
!pip install -q condacolab
import condacolab
condacolab.install()
```

```
 Downloading https://github.com/jaimergp/miniforge/releases/download/24.11.2-
1_colab/Miniforge3-colab-24.11.2-1_colab-Linux-x86_64.sh…
 Installing…
 Adjusting configuration…
 Patching environment…
 Done in 0:00:16
 Restarting kernel…
```

[1]:
```
!wget https://cs.stanford.edu/~minalee/zip/chi2022-coauthor-v1.0.zip
!unzip -q chi2022-coauthor-v1.0.zip
!rm chi2022-coauthor-v1.0.zip
!unzip -q creativity_index-main.zip
```

```
--2025-02-26 18:17:12--
https://cs.stanford.edu/~minalee/zip/chi2022-coauthor-v1.0.zip
Resolving cs.stanford.edu (cs.stanford.edu)… 171.64.64.64
Connecting to cs.stanford.edu (cs.stanford.edu)|171.64.64.64|:443… connected.
HTTP request sent, awaiting response… 200 OK
Length: 49956179 (48M) [application/zip]
Saving to: ‘chi2022-coauthor-v1.0.zip’

chi2022-coauthor-v1 100%[===================>]  47.64M  8.81MB/s    in 5.7s

2025-02-26 18:17:18 (8.34 MB/s) - ‘chi2022-coauthor-v1.0.zip’ saved
[49956179/49956179]

unzip:  cannot find or open creativity_index-main.zip, creativity_index-
main.zip.zip or creativity_index-main.zip.ZIP.
```

[ ]:
```

```

Coauthor cleanup

[1]:
```
import os
```

```
dataset_dir = './coauthor-v1.0'
paths = [
    os.path.join(dataset_dir, path)
    for path in os.listdir(dataset_dir)
    if path.endswith('jsonl')
]

print(f'Successfully downloaded {len(paths)} writing sessions in CoAuthor!')
```

```
Successfully downloaded 1447 writing sessions in CoAuthor!
```

[2]:
```
import json

def read_writing_session(path):
    events = []
    with open(path, 'r') as f:
        for event in f:
            events.append(json.loads(event))
    print(f'Successfully read {len(events)} events in a writing session from␣
 ↪{path}')
    return events
```

[3]:
```
events = read_writing_session(paths[0])
```

```
Successfully read 2405 events in a writing session from
./coauthor-v1.0/608e5b73341a4f3ca937316f99dae32d.jsonl
```

[4]:
```
from collections import defaultdict

def reconstruct_user_text(events):
    user_texts = []
    current_text = ""

    # Sort events by eventNum
    sorted_events = sorted(events, key=lambda e: e["eventNum"])

    for event in sorted_events:
        event_type = event["eventName"]
        text_delta = event["textDelta"]

        if event_type == "text-insert" and text_delta:
            extracted_text = ""
            for op in text_delta['ops']:
                if 'insert' in op:
                    extracted_text += op['insert']
            current_text += extracted_text

        elif event_type == "text-delete" and text_delta:
```

```
            delete_len = 0
            for op in text_delta['ops']:
                if 'delete' in op:
                    delete_len += op['delete']
                elif 'retain' in op:
                    pass
            current_text = current_text[:-delete_len]


        # Store finalized text if user completes a thought
        if event_type == "suggestion-get" or event_type == "suggestion-select":
            if current_text.strip():
                user_texts.append(current_text.strip())
                current_text = ""

    return user_texts

# Process all valid events
user_texts = reconstruct_user_text(events)

print(f"Total reconstructed user-written sentences: {len(user_texts)}")
print("Example user-written sentence:", user_texts)  # Show first 3 examples
```

```
Total reconstructed user-written sentences: 5
Example user-written sentence: ['it is the responsibility of a good citizen to
keep up with the news.', "I think it is important to know what is going on in
the world because it can affect us.  For example, there are many pieces of
legislation that are passed that can have bad current and future affects on the
population.  Due to people not keeping up with the news, the public isn't aware
of the effects of the legislation until it is too late.  There are many instance
where the public iecomes aware of a bad bill and they are able to apply pressure
to their government representatives to make changes.", 'before it is too
late\n\nKeeping up with the news also allows the public to stay informed about
issues such as climate change.', 'We are far behind when it comes to dealing
with climate change.  I believe we would be mounting a better challenge to
climate change if more people were knowledgeable about the effects.  This
knowledge could be garnered from the news.  When you have areas of the world
that are experiencing historical high water levels due to climate change, there
are often news reports being done on these subjects.  If the public is not
paying attention, youhe public will probably not put pressure on their
government representatives to act.', 'I believe that people who go through life
oblivious to the problems that surround them are not good people.  The news
often focuses on problems that may not be a problem for certain individuals--
yet.']
```

[5]:
```
def extract_selected_ai_suggestions(events):
    ai_suggestions = []  # Stores accepted AI-generated suggestions
```

```
    last_suggestion_open = None  # Store the most recent `suggestion-open` event

    for event in events:
        if event["eventName"] == "suggestion-open" and event.
 ↪get("currentSuggestions"):
            last_suggestion_open = event  # Save latest `suggestion-open` event

        elif event["eventName"] == "suggestion-select":
            if last_suggestion_open and last_suggestion_open.
 ↪get("currentSuggestions"):
                selected_index = event.get("currentSuggestionIndex", -1)

                # Ensure selected index is valid
                if 0 <= selected_index <␣
 ↪len(last_suggestion_open["currentSuggestions"]):
                    selected_suggestion =␣
 ↪last_suggestion_open["currentSuggestions"][selected_index]["trimmed"]
                    ai_suggestions.append(selected_suggestion)

    return ai_suggestions

# Process all valid events
ai_suggestions = extract_selected_ai_suggestions(events)

print(f"Total AI Suggestions Accepted: {len(ai_suggestions)}")
print("Example AI Suggestion:", ai_suggestions[:3])  # Show first 3 examples
```

```
Total AI Suggestions Accepted: 2
Example AI Suggestion: ['I think it is important to know what is going on in the
world because it can affect us.', 'For example, the Stop Online Piracy Act
(SOPA) was a bill that would have allowed the government to shut down websites
that were suspected']
```

[6]:
```
def extract_acceptance_status(events):
    """Track if a suggestion was accepted or rejected."""
    acceptance_status = []  # Stores 'accepted' or 'rejected'
    last_suggestion_open = None  # Store the most recent `suggestion-open`

    for event in events:
        if event["eventName"] == "suggestion-open" and event.
 ↪get("currentSuggestions"):
            last_suggestion_open = event  # Save the latest `suggestion-open`

        elif event["eventName"] == "suggestion-select" and last_suggestion_open:
            # Suggestion selected
            acceptance_status.append('accepted')
```

```
            last_suggestion_open = None  # Reset, since we handled the␣
↪acceptance

        elif event["eventName"] == "suggestion-close" and last_suggestion_open:
            # Suggestion closed without selection (rejected)
            acceptance_status.append('rejected')
            last_suggestion_open = None  # Reset, since we handled the rejection

    return acceptance_status
```

[7]:
```
import pandas as pd

jsd_results = []
small = paths[100:110]

def extract_acceptance_status(events):
    """Track if a suggestion was accepted or rejected."""
    acceptance_status = []  # Stores 'accepted' or 'rejected'
    last_suggestion_open = None  # Store the most recent `suggestion-open`

    for event in events:
        if event["eventName"] == "suggestion-open" and event.
↪get("currentSuggestions"):
            last_suggestion_open = event  # Save the latest `suggestion-open`

        elif event["eventName"] == "suggestion-select" and last_suggestion_open:
            # Suggestion selected
            acceptance_status.append('accepted')
            last_suggestion_open = None  # Reset, since we handled the␣
↪acceptance

        elif event["eventName"] == "suggestion-close" and last_suggestion_open:
            # Suggestion closed without selection (rejected)
            acceptance_status.append('rejected')
            last_suggestion_open = None  # Reset, since we handled the rejection

    return acceptance_status
```

[ ]:
```
!conda env create -f environment_infini.yml
!conda env create -f environment_vllm.yml
```

Anil R. Doshi, Oliver P. Hauser ,Generative AI enhances individual creativity but reduces the collective diversity of novel content.Sci. Adv.10,eadn5290(2024).DOI:10.1126/sciadv.adn5290

Doshi and Hauser didn't present a single "magic-formula" but rather defined creativity as a combination of two dimensions—novelty and usefulness—each measured by averaging three constituent ratings. In their study:

- The **novelty index** is computed as the mean of ratings on:

  - **Novelty**
  - **Originality**
  - **Rarity**
- The **usefulness index** is computed as the mean of ratings on:
  - **Appropriateness** (for the target audience)
  - **Feasibility** (of further development)
  - **Publishability** (likelihood a publisher would take it on)

Usefulness is fully captured in the other features of helpfulness.

Inputs and Data: AI-Suggested Text (Candidate)

: The newly generated text from the AI model.

Context Text

: The preceding user text, conversation history, or story background. Used to measure appropriateness and novelty relative to context.

Reference Corpus

A large dataset representing "generic" writing or domain-specific text. Used to measure how common or rare certain phrases, syntactic patterns, or ideas are.

[9]:
```
!pip install transformers
!pip install sacremoses
!pip install numpy
!pip install nltk
```

```
Requirement already satisfied: transformers in /usr/local/lib/python3.11/dist-
packages (4.48.3)
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-
packages (from transformers) (3.17.0)
Requirement already satisfied: huggingface-hub<1.0,>=0.24.0 in
/usr/local/lib/python3.11/dist-packages (from transformers) (0.28.1)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.11/dist-
packages (from transformers) (1.26.4)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.11/dist-packages (from transformers) (24.2)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.11/dist-
packages (from transformers) (6.0.2)
Requirement already satisfied: regex!=2019.12.17 in
/usr/local/lib/python3.11/dist-packages (from transformers) (2024.11.6)
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-
packages (from transformers) (2.32.3)
Requirement already satisfied: tokenizers<0.22,>=0.21 in
/usr/local/lib/python3.11/dist-packages (from transformers) (0.21.0)
Requirement already satisfied: safetensors>=0.4.1 in
/usr/local/lib/python3.11/dist-packages (from transformers) (0.5.2)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.11/dist-
packages (from transformers) (4.67.1)
Requirement already satisfied: fsspec>=2023.5.0 in
```

```
/usr/local/lib/python3.11/dist-packages (from huggingface-
hub<1.0,>=0.24.0->transformers) (2024.10.0)
Requirement already satisfied: typing-extensions>=3.7.4.3 in
/usr/local/lib/python3.11/dist-packages (from huggingface-
hub<1.0,>=0.24.0->transformers) (4.12.2)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.11/dist-packages (from requests->transformers) (3.4.1)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-
packages (from requests->transformers) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.11/dist-packages (from requests->transformers) (2.3.0)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.11/dist-packages (from requests->transformers)
(2025.1.31)
Collecting sacremoses
  Downloading sacremoses-0.1.1-py3-none-any.whl.metadata (8.3 kB)
Requirement already satisfied: regex in /usr/local/lib/python3.11/dist-packages
(from sacremoses) (2024.11.6)
Requirement already satisfied: click in /usr/local/lib/python3.11/dist-packages
(from sacremoses) (8.1.8)
Requirement already satisfied: joblib in /usr/local/lib/python3.11/dist-packages
(from sacremoses) (1.4.2)
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-packages
(from sacremoses) (4.67.1)
Downloading sacremoses-0.1.1-py3-none-any.whl (897 kB)
                         897.5/897.5 kB
8.5 MB/s eta 0:00:00
Installing collected packages: sacremoses
Successfully installed sacremoses-0.1.1
Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-packages
(1.26.4)
Requirement already satisfied: nltk in /usr/local/lib/python3.11/dist-packages
(3.9.1)
Requirement already satisfied: click in /usr/local/lib/python3.11/dist-packages
(from nltk) (8.1.8)
Requirement already satisfied: joblib in /usr/local/lib/python3.11/dist-packages
(from nltk) (1.4.2)
Requirement already satisfied: regex>=2021.8.3 in
/usr/local/lib/python3.11/dist-packages (from nltk) (2024.11.6)
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-packages
(from nltk) (4.67.1)
```

To measure creativity, we'll break this down into two components.

1. Through `DJ_search_exact` by Lu et al. 2024 to find exact matches against a large corpus. We can get how much of the generated suggestions are directly "non-creative." The output is a score from 0-1, 0 if no spans are found, 1 if all tokens up to the last relevant index are matched.
2. Syntactical complexity (later we will measure syntactical similarity).

[10]:
```
!pip install nltk
!conda init
!conda activate infini-gram
!pip install unidecode
```

```
Requirement already satisfied: nltk in /usr/local/lib/python3.11/dist-packages
(3.9.1)
Requirement already satisfied: click in /usr/local/lib/python3.11/dist-packages
(from nltk) (8.1.8)
Requirement already satisfied: joblib in /usr/local/lib/python3.11/dist-packages
(from nltk) (1.4.2)
Requirement already satisfied: regex>=2021.8.3 in
/usr/local/lib/python3.11/dist-packages (from nltk) (2024.11.6)
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-packages
(from nltk) (4.67.1)
/bin/bash: line 1: conda: command not found
/bin/bash: line 1: conda: command not found
Collecting unidecode
  Downloading Unidecode-1.3.8-py3-none-any.whl.metadata (13 kB)
Downloading Unidecode-1.3.8-py3-none-any.whl (235 kB)
                        235.5/235.5 kB
3.5 MB/s eta 0:00:00
Installing collected packages: unidecode
Successfully installed unidecode-1.3.8
```

The following is adapted from Lu et al. 2024.

[19]:
```
import os
import nltk
import json
import time
import requests
import argparse
import numpy as np
from tqdm import tqdm
from typing import List, Callable
from dataclasses import dataclass
from unidecode import unidecode
from sacremoses import MosesDetokenizer
from transformers import AutoTokenizer

md = MosesDetokenizer(lang='en')
API_URL = 'https://api.infini-gram.io/'
HF_TOKEN = "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"
tokenizer = AutoTokenizer.from_pretrained("meta-llama/Llama-2-7b-hf",␣
 ↪token=HF_TOKEN,
                                          add_bos_token=False,␣
 ↪add_eos_token=False)
```

```python
@dataclass
class Document:
    doc_id: str
    tokens: List[str]

@dataclass
class Span:
    start_index: int
    end_index: int
    span_text: str
    occurrence: int

class Hypothesis:
    def __init__(self, target_doc: Document, min_ngram: int) -> None:
        self.target_doc = target_doc
        self.min_ngram = min_ngram
        self.spans = []
        self.finished = False

    def add_span(self, new_span: Span) -> None:
        self.spans.append(new_span)
        if new_span.end_index >= len(self.target_doc.tokens):
            self.finished = True

    def replace_span(self, new_span: Span) -> None:
        self.spans = self.spans[:-1] + [new_span]
        if new_span.end_index >= len(self.target_doc.tokens):
            self.finished = True

    def get_score(self) -> float:
        if not self.spans:
            return 0.0
        progress_len = self.spans[-1].end_index if not self.finished else␣
 ↪len(self.target_doc.tokens)
        flags = [False] * progress_len
        for span in self.spans:
            span_length = span.end_index - span.start_index
            flags[span.start_index:span.end_index] = [True] * span_length
        coverage = sum(flags) / len(flags)
        return coverage

    def format_span(self) -> str:
        return ' | '.join([s.span_text for s in self.spans])

    def __hash__(self) -> int:
        return hash(self.format_span())
```

```
    def __eq__(self, other) -> bool:
        if isinstance(other, Hypothesis):
            return self.format_span() == other.format_span()
        return NotImplemented

    def get_avg_span_len(self) -> float:
        if not self.spans:
            return 0.0
        span_lengths = [s.end_index - s.start_index for s in self.spans]
        return sum(span_lengths) / len(span_lengths)

    def export_json(self) -> dict:
        matched_spans = [{
            'start_index': s.start_index,
            'end_index': s.end_index,
            'span_text': s.span_text,
            'occurrence': s.occurrence
        } for s in self.spans]
        return {
            'matched_spans': matched_spans,
            'coverage': self.get_score(),
            'avg_span_len': self.get_avg_span_len(),
        }

def find_exact_match(detokenize: Callable, doc: Document, min_ngram: int) ->␣
 ↪dict:
    hypothesis = Hypothesis(doc, min_ngram)
    first_pointer, second_pointer = 0, min_ngram
    while second_pointer <= len(doc.tokens):
        span_text = detokenize(doc.tokens[first_pointer:second_pointer])
        request_data = {
            'corpus': 'v4_rpj_llama_s4',
            'engine': 'c++',
            'query_type': 'count',
            'query': span_text,
        }
        time.sleep(0.01)
        search_result = requests.post(API_URL, json=request_data).json()
        occurrence = search_result.get('count', 0)

        if occurrence:
            matched_span = Span(
                start_index=first_pointer,
                end_index=second_pointer,
                span_text=span_text,
                occurrence=occurrence
            )
```

```
            if not hypothesis.spans:
                hypothesis.add_span(matched_span)
            else:
                last_span = hypothesis.spans[-1]
                if matched_span.start_index <= last_span.start_index and␣
↪last_span.end_index <= matched_span.end_index:
                    hypothesis.replace_span(matched_span)
                else:
                    hypothesis.add_span(matched_span)
            second_pointer += 1

            #␣
↪print("**********************************************************************************
            # print(hypothesis.format_span())
            # print(f'score: {hypothesis.get_score():.4f}  avg_span_length:␣
↪{hypothesis.get_avg_span_len()}')
            #␣
↪print("**********************************************************************************
        else:
            if second_pointer - first_pointer > min_ngram:
                first_pointer += 1
            elif second_pointer - first_pointer == min_ngram:
                first_pointer += 1
                second_pointer += 1
            else:
                raise ValueError("Invalid state in span detection.")

    hypothesis.finished = True
    return hypothesis.export_json()

def process_text(text: str, min_ngram: int = 6, lm_tokenizer: bool = False) ->␣
↪dict:
    # Choose the appropriate tokenizer/detokenizer.
    if not lm_tokenizer:
        tokenize_func = lambda x: nltk.tokenize.casual.casual_tokenize(x)
        detokenize = lambda tokens: md.detokenize(tokens)
    else:
        tokenize_func = lambda x: tokenizer.tokenize(x)
        detokenize = lambda tokens: tokenizer.decode(tokenizer.
↪convert_tokens_to_ids(tokens))

    # Preprocess and tokenize the plain text.
    processed_text = unidecode(text)
    tokens = tokenize_func(processed_text)

    if len(tokens) <= min_ngram:
```

```
        raise ValueError("Input text is too short for the specified min_ngram.")

    # Build a Document and perform the exact match search.
    doc = Document(doc_id="input_text", tokens=tokens)
    result = find_exact_match(detokenize, doc, min_ngram)
    return result

print (process_text("I'm the best in the world, laughing out loud.
 ↪")['coverage'])
```

```
1.0
```

[ ]:
```
!pip install spacy
!python -m spacy download en_core_web_sm
```

```
Collecting spacy
  Downloading
spacy-3.8.4-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata
(27 kB)
Collecting spacy-legacy<3.1.0,>=3.0.11 (from spacy)
  Downloading spacy_legacy-3.0.12-py2.py3-none-any.whl.metadata (2.8 kB)
Collecting spacy-loggers<2.0.0,>=1.0.0 (from spacy)
  Downloading spacy_loggers-1.0.5-py3-none-any.whl.metadata (23 kB)
Collecting murmurhash<1.1.0,>=0.28.0 (from spacy)
  Downloading murmurhash-1.0.12-cp311-cp311-
manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_6
4.whl.metadata (2.1 kB)
Collecting cymem<2.1.0,>=2.0.2 (from spacy)
  Downloading
cymem-2.0.11-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata
(8.5 kB)
Collecting preshed<3.1.0,>=3.0.2 (from spacy)
  Downloading preshed-3.0.9-cp311-cp311-
manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_6
4.whl.metadata (2.2 kB)
Collecting thinc<8.4.0,>=8.3.4 (from spacy)
  Downloading
thinc-8.3.4-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata
(15 kB)
Collecting wasabi<1.2.0,>=0.9.1 (from spacy)
  Downloading wasabi-1.1.3-py3-none-any.whl.metadata (28 kB)
Collecting srsly<3.0.0,>=2.4.3 (from spacy)
  Downloading
srsly-2.5.1-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata
(19 kB)
Collecting catalogue<2.1.0,>=2.0.6 (from spacy)
  Downloading catalogue-2.0.10-py3-none-any.whl.metadata (14 kB)
Collecting weasel<0.5.0,>=0.1.0 (from spacy)
```

```
  Downloading weasel-0.4.1-py3-none-any.whl.metadata (4.6 kB)
Collecting typer<1.0.0,>=0.3.0 (from spacy)
  Downloading typer-0.15.1-py3-none-any.whl.metadata (15 kB)
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in
/usr/local/lib/python3.11/site-packages (from spacy) (4.67.1)
Requirement already satisfied: numpy>=1.19.0 in /usr/local/lib/python3.11/site-
packages (from spacy) (2.0.2)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in
/usr/local/lib/python3.11/site-packages (from spacy) (2.32.3)
Collecting pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4 (from spacy)
  Downloading pydantic-2.10.6-py3-none-any.whl.metadata (30 kB)
Collecting jinja2 (from spacy)
  Downloading jinja2-3.1.5-py3-none-any.whl.metadata (2.6 kB)
Requirement already satisfied: setuptools in /usr/local/lib/python3.11/site-
packages (from spacy) (65.6.3)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.11/site-packages (from spacy) (24.2)
Collecting langcodes<4.0.0,>=3.2.0 (from spacy)
  Downloading langcodes-3.5.0-py3-none-any.whl.metadata (29 kB)
Collecting language-data>=1.2 (from langcodes<4.0.0,>=3.2.0->spacy)
  Downloading language_data-1.3.0-py3-none-any.whl.metadata (4.3 kB)
Collecting annotated-types>=0.6.0 (from
pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4->spacy)
  Using cached annotated_types-0.7.0-py3-none-any.whl.metadata (15 kB)
Collecting pydantic-core==2.27.2 (from
pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4->spacy)
  Downloading pydantic_core-2.27.2-cp311-cp311-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (6.6 kB)
Requirement already satisfied: typing-extensions>=4.12.2 in
/usr/local/lib/python3.11/site-packages (from
pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4->spacy) (4.12.2)
Requirement already satisfied: charset_normalizer<4,>=2 in
/usr/local/lib/python3.11/site-packages (from requests<3.0.0,>=2.13.0->spacy)
(3.4.1)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/site-
packages (from requests<3.0.0,>=2.13.0->spacy) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.11/site-packages (from requests<3.0.0,>=2.13.0->spacy)
(2.3.0)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.11/site-packages (from requests<3.0.0,>=2.13.0->spacy)
(2024.12.14)
Collecting blis<1.3.0,>=1.2.0 (from thinc<8.4.0,>=8.3.4->spacy)
  Downloading
blis-1.2.0-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata
(7.7 kB)
Collecting confection<1.0.0,>=0.0.1 (from thinc<8.4.0,>=8.3.4->spacy)
  Downloading confection-0.1.5-py3-none-any.whl.metadata (19 kB)
```

```
Requirement already satisfied: click>=8.0.0 in /usr/local/lib/python3.11/site-
packages (from typer<1.0.0,>=0.3.0->spacy) (8.1.8)
Collecting shellingham>=1.3.0 (from typer<1.0.0,>=0.3.0->spacy)
  Downloading shellingham-1.5.4-py2.py3-none-any.whl.metadata (3.5 kB)
Requirement already satisfied: rich>=10.11.0 in /usr/local/lib/python3.11/site-
packages (from typer<1.0.0,>=0.3.0->spacy) (13.9.4)
Collecting cloudpathlib<1.0.0,>=0.7.0 (from weasel<0.5.0,>=0.1.0->spacy)
  Downloading cloudpathlib-0.20.0-py3-none-any.whl.metadata (14 kB)
Collecting smart-open<8.0.0,>=5.2.1 (from weasel<0.5.0,>=0.1.0->spacy)
  Downloading smart_open-7.1.0-py3-none-any.whl.metadata (24 kB)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.11/site-packages (from jinja2->spacy) (3.0.2)
Collecting marisa-trie>=1.1.0 (from language-
data>=1.2->langcodes<4.0.0,>=3.2.0->spacy)
  Downloading marisa_trie-1.2.1-cp311-cp311-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (9.0 kB)
Requirement already satisfied: markdown-it-py>=2.2.0 in
/usr/local/lib/python3.11/site-packages (from
rich>=10.11.0->typer<1.0.0,>=0.3.0->spacy) (3.0.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in
/usr/local/lib/python3.11/site-packages (from
rich>=10.11.0->typer<1.0.0,>=0.3.0->spacy) (2.19.1)
Requirement already satisfied: wrapt in /usr/local/lib/python3.11/site-packages
(from smart-open<8.0.0,>=5.2.1->weasel<0.5.0,>=0.1.0->spacy) (1.17.2)
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.11/site-
packages (from markdown-it-py>=2.2.0->rich>=10.11.0->typer<1.0.0,>=0.3.0->spacy)
(0.1.2)
Downloading
spacy-3.8.4-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (30.6 MB)
                         30.6/30.6 MB
43.9 MB/s eta 0:00:00
Downloading catalogue-2.0.10-py3-none-any.whl (17 kB)
Downloading
cymem-2.0.11-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (218 kB)
Downloading langcodes-3.5.0-py3-none-any.whl (182 kB)
Downloading murmurhash-1.0.12-cp311-cp311-
manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_6
4.whl (134 kB)
Downloading preshed-3.0.9-cp311-cp311-
manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_6
4.whl (157 kB)
Downloading pydantic-2.10.6-py3-none-any.whl (431 kB)
Downloading
pydantic_core-2.27.2-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
(2.0 MB)
                         2.0/2.0 MB
62.5 MB/s eta 0:00:00
Downloading spacy_legacy-3.0.12-py2.py3-none-any.whl (29 kB)
```

```
Downloading spacy_loggers-1.0.5-py3-none-any.whl (22 kB)
Downloading
srsly-2.5.1-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1.1 MB)
                         1.1/1.1 MB
46.8 MB/s eta 0:00:00
Downloading
thinc-8.3.4-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (3.9 MB)
                         3.9/3.9 MB
85.8 MB/s eta 0:00:00
Downloading typer-0.15.1-py3-none-any.whl (44 kB)
Downloading wasabi-1.1.3-py3-none-any.whl (27 kB)
Downloading weasel-0.4.1-py3-none-any.whl (50 kB)
Downloading jinja2-3.1.5-py3-none-any.whl (134 kB)
Downloading annotated_types-0.7.0-py3-none-any.whl (13 kB)
Downloading
blis-1.2.0-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (11.7 MB)
                         11.7/11.7 MB
72.7 MB/s eta 0:00:00
Downloading cloudpathlib-0.20.0-py3-none-any.whl (52 kB)
Downloading confection-0.1.5-py3-none-any.whl (35 kB)
Downloading language_data-1.3.0-py3-none-any.whl (5.4 MB)
                         5.4/5.4 MB
56.0 MB/s eta 0:00:00
Downloading shellingham-1.5.4-py2.py3-none-any.whl (9.8 kB)
Downloading smart_open-7.1.0-py3-none-any.whl (61 kB)
Downloading
marisa_trie-1.2.1-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
(1.4 MB)
                         1.4/1.4 MB
42.3 MB/s eta 0:00:00
Installing collected packages: cymem, wasabi, spacy-loggers, spacy-legacy,
smart-open, shellingham, pydantic-core, murmurhash, marisa-trie, jinja2,
cloudpathlib, catalogue, blis, annotated-types, srsly, pydantic, preshed,
language-data, typer, langcodes, confection, weasel, thinc, spacy
Successfully installed annotated-types-0.7.0 blis-1.2.0 catalogue-2.0.10
cloudpathlib-0.20.0 confection-0.1.5 cymem-2.0.11 jinja2-3.1.5 langcodes-3.5.0
language-data-1.3.0 marisa-trie-1.2.1 murmurhash-1.0.12 preshed-3.0.9
pydantic-2.10.6 pydantic-core-2.27.2 shellingham-1.5.4 smart-open-7.1.0
spacy-3.8.4 spacy-legacy-3.0.12 spacy-loggers-1.0.5 srsly-2.5.1 thinc-8.3.4
typer-0.15.1 wasabi-1.1.3 weasel-0.4.1
Collecting en-core-web-sm==3.8.0
  Downloading https://github.com/explosion/spacy-
models/releases/download/en_core_web_sm-3.8.0/en_core_web_sm-3.8.0-py3-none-
any.whl (12.8 MB)
                           12.8/12.8 MB
95.6 MB/s eta 0:00:00
Installing collected packages: en-core-web-sm
Successfully installed en-core-web-sm-3.8.0
```

```
 Download and installation successful
You can now load the package via spacy.load('en_core_web_sm')
```

[9]:
```python
import spacy
from collections import deque

# Load spaCy model
nlp = spacy.load("en_core_web_sm")

def get_parse_tree_depth(sent):
    """
    Calculate parse tree depth for a sentence using a BFS
    starting from the root token.
    """
    roots = [token for token in sent if token.head == token]
    if not roots:
        return 0
    root = roots[0]
    max_depth = 0
    depths = {root: 0}
    queue = deque([root])

    while queue:
        node = queue.popleft()
        for child in node.children:
            depths[child] = depths[node] + 1
            max_depth = max(max_depth, depths[child])
            queue.append(child)
    return max_depth

def count_subordinate_clauses(doc):
    """
    Count subordinate clauses by looking for tokens with dependency labels
    typically marking clause boundaries:
      - 'mark' (subordinating conjunctions),
      - 'advcl' (adverbial clauses), and
      - 'ccomp' (clausal complements).
    """
    count = 0
    for token in doc:
        if token.dep_ in {"mark", "advcl", "ccomp"}:
            count += 1
    return count

def count_passive_sentences(doc):
    """
    Identify and count passive sentences. A sentence is considered passive
```

```
    if it contains any token with the dependency 'nsubjpass'.
    """
    passive_count = 0
    for sent in doc.sents:
        if any(token.dep_ == "nsubjpass" for token in sent):
            passive_count += 1
    return passive_count

def syntactic_complexity(text, d_max=10, max_subordinate_per_sentence=2):
    """
    Calculate a normalized syntactic complexity score in the range [0,1].
    It is very knowledge-based.
    This function computes three components:
      1. Depth Component: Average parse tree depth normalized by d_max.
         The average is capped at d_max to avoid outlier effects.
      2. Subordinate Clause Component: The average number of subordinate clauses
         per sentence normalized by an assumed maximum (default=2).
      3. Passive Voice Component: The fraction of sentences exhibiting passive␣
 ↪voice.

    The final score is the average of these three components.
    """
    doc = nlp(text)
    sentences = list(doc.sents)
    num_sentences = len(sentences)

    if num_sentences == 0:
        return 0.0

    # Depth Component
    total_depth = sum(get_parse_tree_depth(sent) for sent in sentences)
    avg_depth = total_depth / num_sentences
    depth_component = min(avg_depth, d_max) / d_max  # normalize to [0,1]

    # Subordinate Clause Component
    num_subordinate = count_subordinate_clauses(doc)
    subordinate_component = (num_subordinate / num_sentences) /␣
 ↪max_subordinate_per_sentence
    subordinate_component = min(subordinate_component, 1.0)

    # Passive Voice Component
    passive_sentences = count_passive_sentences(doc)
    passive_component = passive_sentences / num_sentences  # already in [0,1]

    # Combine components equally to yield a final score between 0 and 1
    normalized_score = (depth_component + subordinate_component +␣
 ↪passive_component) / 3.0
```

```
    return normalized_score

text = (
    "The old mansion, which had been abandoned for years, stood silent. Its␣
 ↪walls, marked by time and neglect, told stories as if they were whispering.␣
 ↪The garden was overgrown, and nature had reclaimed it, making it look like a␣
 ↪scene from a forgotten fairy tale."
)

score = syntactic_complexity(text)
print("Normalized Syntactic Complexity Score:", score)
```

```
Normalized Syntactic Complexity Score: 0.5777777777777777
```

[ ]:

[10]:
```
def extract_acceptance_status(events):
    """Track if a suggestion was accepted or rejected."""
    acceptance_status = []  # Stores 'accepted' or 'rejected'
    last_suggestion_open = None  # Store the most recent `suggestion-open`

    for event in events:
        if event["eventName"] == "suggestion-open" and event.
 ↪get("currentSuggestions"):
            last_suggestion_open = event  # Save the latest `suggestion-open`

        elif event["eventName"] == "suggestion-select" and last_suggestion_open:
            # Suggestion selected
            acceptance_status.append('accepted')
            last_suggestion_open = None  # Reset, since we handled the␣
 ↪acceptance

        elif event["eventName"] == "suggestion-close" and last_suggestion_open:
            # Suggestion closed without selection (rejected)
            acceptance_status.append('rejected')
            last_suggestion_open = None  # Reset, since we handled the rejection

    return acceptance_status
```

[20]:
```
# Loop through each path (limited to paths 100)
jsd_results = []
for path in paths[:100]:
    events = read_writing_session(path)

    if len(events) < 3000:

      # Extract user-written text and AI-generated suggestions
```

```
    user_texts = reconstruct_user_text(events)
    ai_suggestions = extract_selected_ai_suggestions(events)
    acceptance_status = extract_acceptance_status(events)

    # Ensure we only compare matching pairs
    for user_text, ai_text, status in zip(user_texts, ai_suggestions,␣
↪acceptance_status):
        if len(ai_text.split()) < 6:
            continue
        user_score = syntactic_complexity(user_text)
        ai_score = syntactic_complexity(ai_text)
        coverage = process_text(ai_text)['coverage']
        # Store results, including acceptance status
        jsd_results.append({
            "path": path,
            "user_text": user_text,
            "ai_suggestion": ai_text,
            "acceptance_status": status,
            "user_score": user_score,
            "ai_score": ai_score,
            "coverage": coverage
        })
```

```
Successfully read 2405 events in a writing session from
./coauthor-v1.0/608e5b73341a4f3ca937316f99dae32d.jsonl
Successfully read 2830 events in a writing session from
./coauthor-v1.0/2d9358657a364ccab47e9a7538cb6650.jsonl
Successfully read 1075 events in a writing session from
./coauthor-v1.0/8c11358444974bf0b5224183acd8149d.jsonl
Successfully read 1377 events in a writing session from
./coauthor-v1.0/ea10c484cd6245f6a5840eea4b4b143b.jsonl
Successfully read 2381 events in a writing session from
./coauthor-v1.0/87396211fee244ccbda4b0f4bcefc8e9.jsonl
Successfully read 1312 events in a writing session from
./coauthor-v1.0/f4d98feb9e2f478abc3817bdfef5cca0.jsonl
Successfully read 1968 events in a writing session from
./coauthor-v1.0/6938b4a468aa482ba459d2fbe140cb30.jsonl
Successfully read 2910 events in a writing session from
./coauthor-v1.0/b609372842a94db2b278d4ad7f7e366b.jsonl
Successfully read 2329 events in a writing session from
./coauthor-v1.0/7b7b1b40027c4974ade1850596e6726f.jsonl
Successfully read 2250 events in a writing session from
./coauthor-v1.0/ee15d49584a54e8ab79b140e400276ea.jsonl
Successfully read 2262 events in a writing session from
./coauthor-v1.0/ba6d32a6c29a4bfeacf1e25a87a1f804.jsonl
Successfully read 1615 events in a writing session from
./coauthor-v1.0/0ff12af0651e4b0eaf88e22e82058137.jsonl
Successfully read 863 events in a writing session from
```

./coauthor-v1.0/9c4942cdf0d94b6d9ce49896e0093da6.jsonl
Successfully read 1803 events in a writing session from
./coauthor-v1.0/09770cd9061f456fa6d82db00b43a0bd.jsonl
Successfully read 2178 events in a writing session from
./coauthor-v1.0/9ec10cba9d8c48428d656fceff57e834.jsonl
Successfully read 2235 events in a writing session from
./coauthor-v1.0/e7107e7319eb4ab18a7e3fc4445b2d91.jsonl
Successfully read 2947 events in a writing session from
./coauthor-v1.0/eec20f24484d418eb4ba564fbfbc2ab6.jsonl
Successfully read 2805 events in a writing session from
./coauthor-v1.0/6d6a8d63190243f685256031a59166dd.jsonl
Successfully read 1357 events in a writing session from
./coauthor-v1.0/a87286598bc54313b39f6ce6013541bf.jsonl
Successfully read 1626 events in a writing session from
./coauthor-v1.0/a215d4b3242a4e999ee6fafdd793cac6.jsonl
Successfully read 2433 events in a writing session from
./coauthor-v1.0/79a6126076714a1ba4fc9abc52c9201d.jsonl
Successfully read 2397 events in a writing session from
./coauthor-v1.0/f0b4803428a846c6b22e2d420e980a33.jsonl
Successfully read 1516 events in a writing session from
./coauthor-v1.0/1ec7d03923814ad590d270c260b37fad.jsonl
Successfully read 1194 events in a writing session from
./coauthor-v1.0/e29599231e66474b8a524c23db775e47.jsonl
Successfully read 3366 events in a writing session from
./coauthor-v1.0/0a83297c1ce94d15b358d79e533396c7.jsonl
Successfully read 1078 events in a writing session from
./coauthor-v1.0/6fb96c88e7b942e6b225baa611dd9738.jsonl
Successfully read 1877 events in a writing session from
./coauthor-v1.0/0bcb6a7180d64d7c9af732f3ad5aff51.jsonl
Successfully read 2686 events in a writing session from
./coauthor-v1.0/3be583425f4c4656a906a036b13e963e.jsonl
Successfully read 977 events in a writing session from
./coauthor-v1.0/d4b47a8a66b84e8ebdf986e6d4edf139.jsonl
Successfully read 1809 events in a writing session from
./coauthor-v1.0/3959b06dd9344c6aa484b3d6b14b4a27.jsonl
Successfully read 1966 events in a writing session from
./coauthor-v1.0/c1714313080c448d857756a2e0ad6071.jsonl
Successfully read 1537 events in a writing session from
./coauthor-v1.0/ace79657baa04aaca50a076720ee1fe8.jsonl
Successfully read 1021 events in a writing session from
./coauthor-v1.0/36bc101319da4b3590b96bce76f7c02c.jsonl
Successfully read 2355 events in a writing session from
./coauthor-v1.0/8393a8ed335f438c9ad5ca3d25a947b6.jsonl
Successfully read 2155 events in a writing session from
./coauthor-v1.0/b768132809f240b2a3efa1719be82ae4.jsonl
Successfully read 1033 events in a writing session from
./coauthor-v1.0/e2cdf3ffc1fb4396a67368c8331d4f58.jsonl
Successfully read 1830 events in a writing session from

./coauthor-v1.0/423daa7fb2214d1093f73d4966240646.jsonl
Successfully read 2828 events in a writing session from
./coauthor-v1.0/d714bc54e0dc4819bafa77a93927817b.jsonl
Successfully read 2573 events in a writing session from
./coauthor-v1.0/105bf88bb4bc42688e06a54644e2989b.jsonl
Successfully read 2257 events in a writing session from
./coauthor-v1.0/09644506674341f5b14e0d62c88ee70a.jsonl
Successfully read 1902 events in a writing session from
./coauthor-v1.0/b30082b36e764dbd859c020b10fe78b5.jsonl
Successfully read 1361 events in a writing session from
./coauthor-v1.0/8d6e9dac68324fae884571881f5b21f1.jsonl
Successfully read 2672 events in a writing session from
./coauthor-v1.0/25742b852b764843b15266008afdc15d.jsonl
Successfully read 2426 events in a writing session from
./coauthor-v1.0/b65e8358d12a498fa442f3d1072c4bb2.jsonl
Successfully read 1266 events in a writing session from
./coauthor-v1.0/de90bdd73b72426daacc29cd1e1a2e97.jsonl
Successfully read 1399 events in a writing session from
./coauthor-v1.0/b75ef56faed74be0b4b59e15138e18f4.jsonl
Successfully read 1603 events in a writing session from
./coauthor-v1.0/adb2b2bbd3fb48fa8433ee84a26949a5.jsonl
Successfully read 1621 events in a writing session from
./coauthor-v1.0/3cfab79e841a41a7b6edf77e7b28886b.jsonl
Successfully read 1851 events in a writing session from
./coauthor-v1.0/e8cdd0418ddb4263ae2799ded6316693.jsonl
Successfully read 2560 events in a writing session from
./coauthor-v1.0/5773114053f147a69bd6790843ea4fd8.jsonl
Successfully read 1590 events in a writing session from
./coauthor-v1.0/d374cad6be364ca08df8064807b4696e.jsonl
Successfully read 3397 events in a writing session from
./coauthor-v1.0/c326ab6f04ca475a93d6322122ee6eb0.jsonl
Successfully read 2085 events in a writing session from
./coauthor-v1.0/2194c0e020934c74b8ab6c28d758feb2.jsonl
Successfully read 1968 events in a writing session from
./coauthor-v1.0/3969517100834dbc813b628a27fed790.jsonl
Successfully read 1498 events in a writing session from
./coauthor-v1.0/749e032914384e4da8f2392a19066b89.jsonl
Successfully read 2062 events in a writing session from
./coauthor-v1.0/876a292223a74b1a8ad5525d93b26d69.jsonl
Successfully read 708 events in a writing session from
./coauthor-v1.0/8d5bc658213f46b881b2a862bfd5a105.jsonl
Successfully read 1787 events in a writing session from
./coauthor-v1.0/f84ff332dc06486c9d61698be8666941.jsonl
Successfully read 1733 events in a writing session from
./coauthor-v1.0/c0bda7b135404137b66b5ade3bf1c8bc.jsonl
Successfully read 805 events in a writing session from
./coauthor-v1.0/8637ce05b848420f920d5ca02f88a89a.jsonl
Successfully read 1839 events in a writing session from

```
./coauthor-v1.0/29056a31430b414582f488535aa0fe91.jsonl
Successfully read 1476 events in a writing session from
./coauthor-v1.0/409ea6cd2f8347dbb41344d1243c7e17.jsonl
Successfully read 2005 events in a writing session from
./coauthor-v1.0/d4a257392aa84174be6166c9c664caae.jsonl
Successfully read 1227 events in a writing session from
./coauthor-v1.0/9e408240b6c34495b3d20597eaa1dd80.jsonl
Successfully read 2045 events in a writing session from
./coauthor-v1.0/09c68920f9e5434bb35a22e1909f324a.jsonl
Successfully read 1502 events in a writing session from
./coauthor-v1.0/4efb9d8419ee42a3b2ec99f01f94e0a6.jsonl
Successfully read 1247 events in a writing session from
./coauthor-v1.0/08b1a2012f4d4115b56cc8231449ff48.jsonl
Successfully read 1432 events in a writing session from
./coauthor-v1.0/66b3ac872822426e81bab531fb15d8a0.jsonl
Successfully read 2619 events in a writing session from
./coauthor-v1.0/5c1905e6a05c4203b624c4db10ede78a.jsonl
Successfully read 2564 events in a writing session from
./coauthor-v1.0/dcb51651d76c424ebd58a8802629f05c.jsonl
Successfully read 1925 events in a writing session from
./coauthor-v1.0/7e400fae43e543b0a6719cc2974029de.jsonl
Successfully read 955 events in a writing session from
./coauthor-v1.0/1059f00ce0f042808d27139c2a2ca9c5.jsonl
Successfully read 1757 events in a writing session from
./coauthor-v1.0/f6c2fa4230df43b9b9857e71b9d68a89.jsonl
Successfully read 1377 events in a writing session from
./coauthor-v1.0/931c030acd5e41b1bfac856da1aeb8e3.jsonl
Successfully read 2762 events in a writing session from
./coauthor-v1.0/13b2ae3d425e4b76b887182418ba4f58.jsonl
Successfully read 1275 events in a writing session from
./coauthor-v1.0/5f020c0bdea94db5aad0478ba02f2924.jsonl
Successfully read 1494 events in a writing session from
./coauthor-v1.0/f151863fbea2417c9bbec7082abaefd6.jsonl
Successfully read 1776 events in a writing session from
./coauthor-v1.0/40d132e1bc0146c0920d7b38c5b9f7c7.jsonl
Successfully read 1893 events in a writing session from
./coauthor-v1.0/fe677c513394449e987eb3ccce62d03c.jsonl
Successfully read 1355 events in a writing session from
./coauthor-v1.0/fbd087c5c3ab48f1b6540c80976e3285.jsonl
Successfully read 2196 events in a writing session from
./coauthor-v1.0/0a08f7fd748246c8ad4e353edb7e2394.jsonl
Successfully read 3177 events in a writing session from
./coauthor-v1.0/4d06b9aa68764d749ef6fffb4fcb54f4.jsonl
Successfully read 1805 events in a writing session from
./coauthor-v1.0/65936cbe394149c6b24802cad630e2d7.jsonl
Successfully read 975 events in a writing session from
./coauthor-v1.0/c790c4b915e84dbf8e283b598105ea90.jsonl
Successfully read 1725 events in a writing session from
```

```
./coauthor-v1.0/4181c7d35eff4eceb4aad545dd2a9273.jsonl
Successfully read 1190 events in a writing session from
./coauthor-v1.0/787e032a5802459e9ac5731cdeb7b077.jsonl
Successfully read 1243 events in a writing session from
./coauthor-v1.0/a23600c207bd470c929852b33b9a2946.jsonl
Successfully read 1259 events in a writing session from
./coauthor-v1.0/864b77f0d5f34f2b9946e518cdeee98a.jsonl
Successfully read 3376 events in a writing session from
./coauthor-v1.0/f3ecf0ce872940238b0bd429a3e36322.jsonl
Successfully read 460 events in a writing session from
./coauthor-v1.0/1524055efcec4b43b192c6b706e44b1e.jsonl
Successfully read 935 events in a writing session from
./coauthor-v1.0/b15bdf5fdb764c3b97f9ec4ab3d348e3.jsonl
Successfully read 1986 events in a writing session from
./coauthor-v1.0/36eead66360847afba6dc9ebf6d99593.jsonl
Successfully read 2125 events in a writing session from
./coauthor-v1.0/3c2459fc295b4836b027ff004c6b88c1.jsonl
Successfully read 3457 events in a writing session from
./coauthor-v1.0/5de04cda76194f70ba4a49ba5c34ad14.jsonl
Successfully read 1729 events in a writing session from
./coauthor-v1.0/a9aa499623e3453dbcd52de1c862f71b.jsonl
Successfully read 2966 events in a writing session from
./coauthor-v1.0/64e545e776c1466687f0fe0dc8bb0d88.jsonl
Successfully read 1309 events in a writing session from
./coauthor-v1.0/947065accb8b402c98fb2aa62a497842.jsonl
Successfully read 3460 events in a writing session from
./coauthor-v1.0/4bdb5a30d17d40d7b160d942552c190b.jsonl
Successfully read 1689 events in a writing session from
./coauthor-v1.0/c95024d4d29d4d90a8aa665aef6397aa.jsonl
Successfully read 1540 events in a writing session from
./coauthor-v1.0/ff21eabc9606411ba8cd583fdcdfe72c.jsonl
```

[21]:
```python
df_results_with_acceptances = pd.DataFrame(jsd_results)
df_results_with_acceptances.to_csv("creativity1.csv", index=False)

print(f"Processed. Results saved to creativity1.csv.")
```

```
Processed. Results saved to creativity1.csv.
```

show 5 highest syntactical complexity.

[22]:
```python
highest_5 = df_results_with_acceptances[df_results_with_acceptances['ai_score']␣
↪< 1].nlargest(5, 'ai_score')
lowest_5 = df_results_with_acceptances.nsmallest(5, 'ai_score')
print("Highest 5 AI Coverage:")
for index, row in highest_5.iterrows():
    print(f"Score: {row['ai_score']}")
    print(f"User Text: {row['user_text']}")
```

```
    print(f"AI Suggestion: {row['ai_suggestion']}")


print("\nLowest 5 AI Coverage:")
for index, row in lowest_5.iterrows():
    print(f"Score: {row['ai_score']}")
    print(f"User Text: {row['user_text']}")
    print(f"AI Suggestion: {row['ai_suggestion']}")
highest_5cover =␣
 ↪df_results_with_acceptances[df_results_with_acceptances['coverage'] < 1].
 ↪nlargest(5, 'coverage')
lowest_5cover= df_results_with_acceptances.nsmallest(5, 'coverage')
print("Highest 5 coverages:")
for index, row in highest_5.iterrows():
    print(f"Score: {row['coverage']}")
    print(f"User Text: {row['user_text']}")
    print(f"AI Suggestion: {row['ai_suggestion']}")

print("lowest 5 coverages:")
for index, row in lowest_5cover.iterrows():
    print(f"Score: {row['coverage']}")
    print(f"User Text: {row['user_text']}")
    print(f"AI Suggestion: {row['ai_suggestion']}")
```

```
Highest 5 Similarity Scores:
Score: 0.9666666666666667
User Text: Obama gasps and cannot believe what he sees.
AI Suggestion: JFK goes on to tell Obama that the society is governed by a group
of twelve men (all former presidents), he is one of them
Score: 0.9
User Text: that they're just too much of it, with the news being reported daily,
is that the news has become more biased than ever. A person that is not keeping
up with the news is more responsible and more
AI Suggestion: inclined to be a well-rounded citizen, but what's happening now
is that uninformed citizens are being taken advantage of by the media and the
government
Score: 0.9
User Text: I do not believe that schools should provide free pads and tampons
for students.  The school should not have to worry about feminine products for
the students.  This should be the responsiblity of the guardiansiparental
student's.
AI Suggestion: I believe that pads and tampons should be treated like toilet
paper, soap, band-aids and other products that are already provided
Score: 0.9
User Text: The shapeshifter himself has always been very talented, and is a
doctor by trade. He has decided to shape shift into a more handsome doctor, kind
of like a George Clooney from ER type, because he knows she loves the show. The
shapeshifter, Jim, has done his research and found out that the woman, Karen,
```

goes to her primary care physician at the Coastal Medical institute.
AI Suggestion: He also knows that she's been going to Dr. John, who is a specialist, for her asthma.
Score: 0.9
User Text: Two men with black suits come up in a black car and tell Obama to get in.
AI Suggestion: He tells him that he was not the only president who was apart of the society, but since his retirement, Obama is now being asked to join.

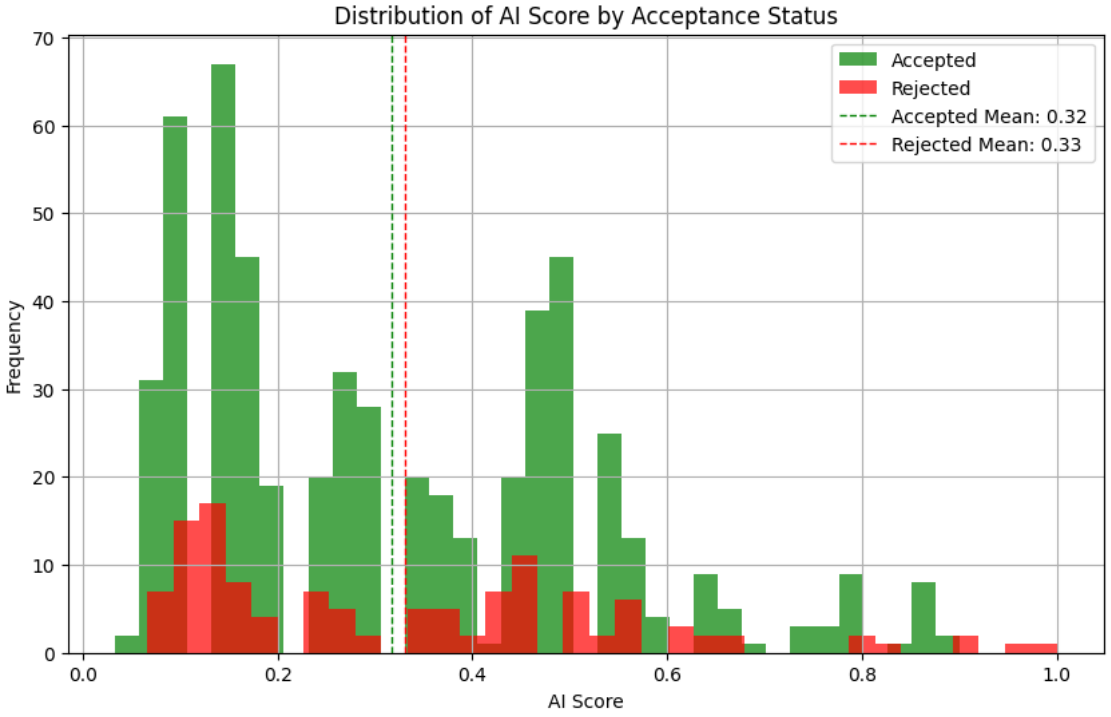
Lowest 5 Similarity Scores:
Score: 0.03333333333333333
User Text: "Mom, I've been trying to get a girl to like me, but nothing seems to wor"
AI Suggestion: Why don't you just be yourself?"
Score: 0.044444444444444446
User Text: "Oh, thank goodness you're back!" says Matt Damon. "How did you do that? And where did you go?"
AI Suggestion: "Oh, thank goodness you're back!" says Matt Damon.
Score: 0.06666666666666667
User Text: From there it was a mad dash back to the bunker, followed by days of hushed talk and whispered rumors.
 The kid could do things now, but he had almost no control over them. And peopel started talking about the old books, how all the little tricks people could do - light a candle, make a gift invisible, etc. - used to be real poeople So other peopel started touching the stone,.
AI Suggestion: And they started feeling things too.
Score: 0.06666666666666667
User Text: and they will check it to see if they have missed anything.sneak away
The pandemic has made thi worse,
AI Suggestion: , are not a good thing.
Score: 0.06666666666666667
User Text: DAD #1 DAD #2 DAD.
AI Suggestion: DAD #1 DAD #2 DAD #3
Highest 5 coverages:
Score: 0.7142857142857143
User Text: Obama gasps and cannot believe what he sees.
AI Suggestion: JFK goes on to tell Obama that the society is governed by a group of twelve men (all former presidents), he is one of them
Score: 0.9230769230769231
User Text: that they're just too much of it, with the news being reported daily, is that the news has become more biased than ever. A person that is not keeping up with the news is more responsible and more
AI Suggestion: inclined to be a well-rounded citizen, but what's happening now is that uninformed citizens are being taken advantage of by the media and the government
Score: 0.8260869565217391
User Text: I do not believe that schools should provide free pads and tampons for students. The school should not have to worry about feminine products for

the students. This should be the responsiblity of the guardiansiparental student's.
AI Suggestion: I believe that pads and tampons should be treated like toilet paper, soap, band-aids and other products that are already provided
Score: 0.5714285714285714
User Text: The shapeshifter himself has always been very talented, and is a doctor by trade. He has decided to shape shift into a more handsome doctor, kind of like a George Clooney from ER type, because he knows she loves the show. The shapeshifter, Jim, has done his research and found out that the woman, Karen, goes to her primary care physician at the Coastal Medical institute.
AI Suggestion: He also knows that she's been going to Dr. John, who is a specialist, for her asthma.
Score: 0.8333333333333334
User Text: Two men with black suits come up in a black car and tell Obama to get in.
AI Suggestion: He tells him that he was not the only president who was apart of the society, but since his retirement, Obama is now being asked to join.
lowest 5 coverages:
Score: 0.0
User Text: The steady hum of the power generators, the quiet conversation of her fellow survivors, and a familiar loping thump as Herman ran by outside. It seemed like just any other day. But she knew it wasn't. It was time.
AI Suggestion: With a heavy sigh, Morgan pushed herself up.
Score: 0.0
User Text: From there it was a mad dash back to the bunker, followed by days of hushed talk and whispered rumors.
 The kid could do things now, but he had almost no control over them. And peopel started talking about the old books, how all the little tricks people could do - light a candle, make a gift invisible, etc. - used to be real poeople So other peopel started touching the stone,.
AI Suggestion: And they started feeling things too.
Score: 0.0
User Text: But it what way? Before…she was always a bit strongr with her trickse, almost powerful by t standards.
AI Suggestion: Was she about to become…something else?
Score: 0.0
User Text: Whatever the reason, Morgan was grateful not to have to worry about any other distractions. SHe ignored the one observer in the room, and walked straight to the stone.
AI Suggestion: as her hand connected, and a pulse of light flowed out.
Score: 0.0
User Text: it seems like people portray characters of Asian descent offensively far too often.
AI Suggestion: Asian women are portrayed as either a submissive exotic beauty or as an overbearing femme fatale.

[23]:
```
import matplotlib.pyplot as plt
# Statistical analysis of ai_score
ai_score_stats = df_results_with_acceptances['ai_score'].describe()
print("AI Score Statistics:")
print(ai_score_stats)

# Group ai_score by acceptance status
ai_score_by_acceptance = df_results_with_acceptances.
 ↪groupby('acceptance_status')['ai_score'].describe()
print("\nAI Score by Acceptance Status:")
print(ai_score_by_acceptance)

# Plotting ai_score distribution by acceptance status (enhanced)
plt.figure(figsize=(10, 6))
plt.
 ↪hist(df_results_with_acceptances[df_results_with_acceptances['acceptance_status']␣
 ↪== 'accepted']['ai_score'],
        bins=35, color='green', alpha=0.7, label='Accepted')
plt.
 ↪hist(df_results_with_acceptances[df_results_with_acceptances['acceptance_status']␣
 ↪== 'rejected']['ai_score'],
        bins=35, color='red', alpha=0.7, label='Rejected')

# Add descriptive statistics to the plot
accepted_ai_mean =␣
 ↪df_results_with_acceptances[df_results_with_acceptances['acceptance_status']␣
 ↪== 'accepted']['ai_score'].mean()
rejected_ai_mean =␣
 ↪df_results_with_acceptances[df_results_with_acceptances['acceptance_status']␣
 ↪== 'rejected']['ai_score'].mean()
plt.axvline(accepted_ai_mean, color='green', linestyle='dashed', linewidth=1,
            label=f'Accepted Mean: {accepted_ai_mean:.2f}')
plt.axvline(rejected_ai_mean, color='red', linestyle='dashed', linewidth=1,
            label=f'Rejected Mean: {rejected_ai_mean:.2f}')

plt.title('Distribution of AI Score by Acceptance Status')
plt.xlabel('AI Score')
plt.ylabel('Frequency')
plt.legend()
plt.grid(True)  # add gridlines for better readability
plt.show()
```

```
AI Score Statistics:
count    668.000000
mean       0.319428
std        0.205482
min        0.033333
```

```
25%         0.133333
50%         0.266667
75%         0.466667
max         1.000000
Name: ai_score, dtype: float64

AI Score by Acceptance Status:
                    count      mean       std       min       25%       50%  \
acceptance_status
accepted            544.0  0.316932  0.201711  0.033333  0.133333  0.266667
rejected            124.0  0.330376  0.221802  0.066667  0.133333  0.266667

                         75%  max
acceptance_status
accepted            0.466667  0.9
rejected            0.466667  1.0
```

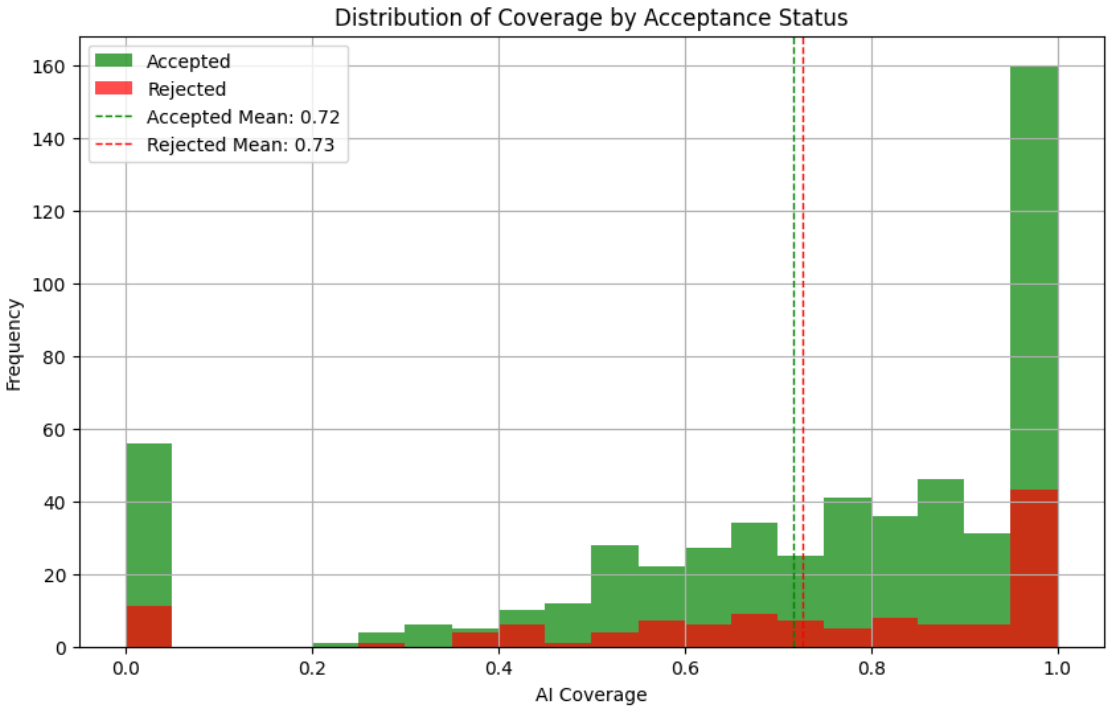


[24]:
```python
# Statistical analysis of coverage
coverage_stats = df_results_with_acceptances['coverage'].describe()
print("Coverage Statistics:")
print(coverage_stats)

# Group coverage by acceptance status
```

```
coverage_by_acceptance = df_results_with_acceptances.
 ↪groupby('acceptance_status')['coverage'].describe()
print("\nCoverage by Acceptance Status:")
print(coverage_by_acceptance)


# Plotting coverage distribution by acceptance status (enhanced)
plt.figure(figsize=(10, 6))
plt.
 ↪hist(df_results_with_acceptances[df_results_with_acceptances['acceptance_status']␣
 ↪== 'accepted']['coverage'], bins=20, color='green', alpha=0.7,␣
 ↪label='Accepted')
plt.
 ↪hist(df_results_with_acceptances[df_results_with_acceptances['acceptance_status']␣
 ↪== 'rejected']['coverage'], bins=20, color='red', alpha=0.7,␣
 ↪label='Rejected')

# Add descriptive statistics to the plot
accepted_mean =␣
 ↪df_results_with_acceptances[df_results_with_acceptances['acceptance_status']␣
 ↪== 'accepted']['coverage'].mean()
rejected_mean =␣
 ↪df_results_with_acceptances[df_results_with_acceptances['acceptance_status']␣
 ↪== 'rejected']['coverage'].mean()
plt.axvline(accepted_mean, color='green', linestyle='dashed', linewidth=1,␣
 ↪label=f'Accepted Mean: {accepted_mean:.2f}')
plt.axvline(rejected_mean, color='red', linestyle='dashed', linewidth=1,␣
 ↪label=f'Rejected Mean: {rejected_mean:.2f}')

plt.title('Distribution of Coverage by Acceptance Status')
plt.xlabel('AI Coverage')
plt.ylabel('Frequency')
plt.legend()
plt.grid(True) # add gridlines for better readability
plt.show()


# Correlation analysis (example)
correlation = df_results_with_acceptances['coverage'].
 ↪corr(df_results_with_acceptances['ai_score'])
print(f"\nCorrelation between Coverage and AI Score: {correlation}")
```

```
Coverage Statistics:
count    668.000000
mean       0.717964
std        0.303438
min        0.000000
```

```
25%         0.587010
50%         0.800000
75%         1.000000
max         1.000000
Name: coverage, dtype: float64

Coverage by Acceptance Status:
                    count      mean       std  min       25%  50%  75%  max
acceptance_status
accepted            544.0  0.716087  0.304044  0.0  0.587010  0.8  1.0  1.0
rejected            124.0  0.726199  0.301852  0.0  0.595833  0.8  1.0  1.0
```




```
Correlation between Coverage and AI Score: 0.11951377105708061
```

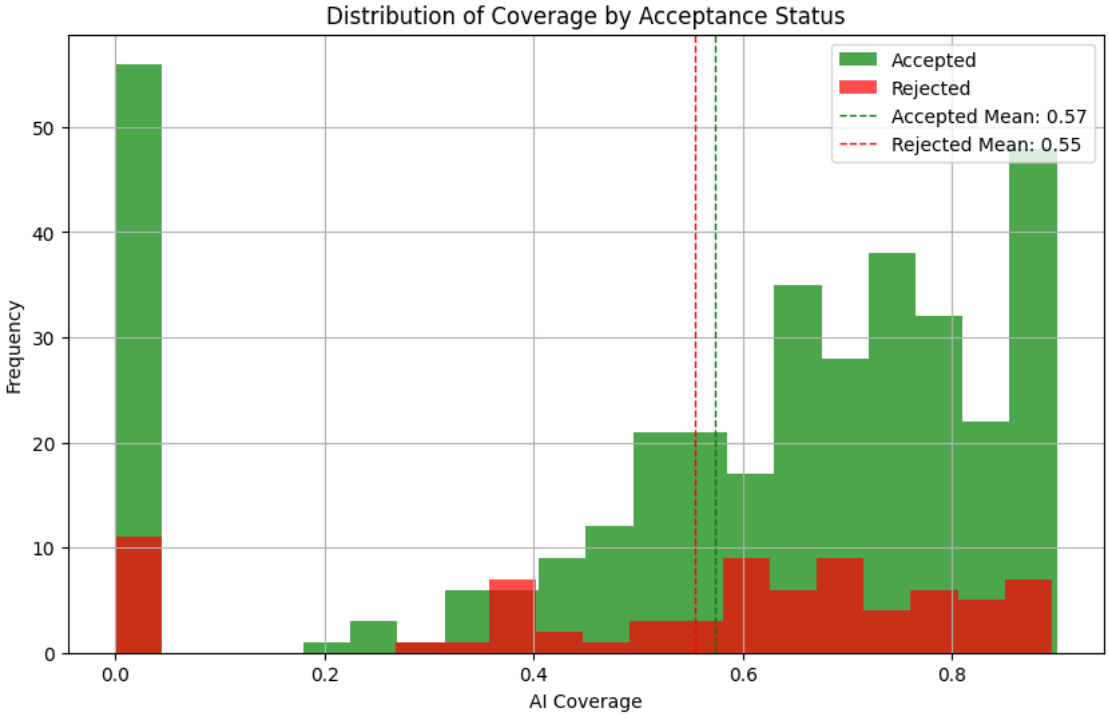
too high coverage might be not good for analysis

[25]:
```
# Filter out rows where coverage > 0.9
df_filtered =␣
  ↪df_results_with_acceptances[df_results_with_acceptances['coverage'] <= 0.9]

# Statistical analysis of coverage
coverage_stats = df_filtered['coverage'].describe()
print("Coverage Statistics:")
print(coverage_stats)
```

```python
# Group coverage by acceptance status
coverage_by_acceptance = df_filtered.groupby('acceptance_status')['coverage'].
 ↪describe()
print("\nCoverage by Acceptance Status:")
print(coverage_by_acceptance)

# Plotting coverage distribution by acceptance status (enhanced)
plt.figure(figsize=(10, 6))
plt.hist(df_filtered[df_filtered['acceptance_status'] ==␣
 ↪'accepted']['coverage'], bins=20, color='green', alpha=0.7, label='Accepted')
plt.hist(df_filtered[df_filtered['acceptance_status'] ==␣
 ↪'rejected']['coverage'], bins=20, color='red', alpha=0.7, label='Rejected')

# Add descriptive statistics to the plot
accepted_mean = df_filtered[df_filtered['acceptance_status'] ==␣
 ↪'accepted']['coverage'].mean()
rejected_mean = df_filtered[df_filtered['acceptance_status'] ==␣
 ↪'rejected']['coverage'].mean()
plt.axvline(accepted_mean, color='green', linestyle='dashed', linewidth=1,␣
 ↪label=f'Accepted Mean: {accepted_mean:.2f}')
plt.axvline(rejected_mean, color='red', linestyle='dashed', linewidth=1,␣
 ↪label=f'Rejected Mean: {rejected_mean:.2f}')

plt.title('Distribution of Coverage by Acceptance Status')
plt.xlabel('AI Coverage')
plt.ylabel('Frequency')
plt.legend()
plt.grid(True)
plt.show()

# Correlation analysis (example)
correlation = df_filtered['coverage'].corr(df_filtered['ai_score'])
print(f"\nCorrelation between Coverage and AI Score: {correlation}")
```

```
Coverage Statistics:
count    431.000000
mean       0.569908
std        0.283726
min        0.000000
25%        0.461538
50%        0.666667
75%        0.777778
max        0.900000
Name: coverage, dtype: float64

Coverage by Acceptance Status:
```

```
                   count      mean       std  min       25%       50%  \
acceptance_status
accepted           356.0  0.573186  0.285929  0.0  0.465385  0.666667
rejected            75.0  0.554352  0.274353  0.0  0.400000  0.625000

                        75%       max
acceptance_status
accepted           0.783385  0.900000
rejected           0.743421  0.894737
```




```
Correlation between Coverage and AI Score: 0.21174872318229554
```