NLP Homework 01: Corpus Analysis
Winter 2026 Semester

Name: Merve Filiz Baker
Date: 02.16.2026

1. Dataset

1.1 Dataset Description and Motivation

For this project, I chose to analyze movie scripts from two distinct genres: Romantic and Sci-Fi. I collected 10 films total(5 from each category)from The Internet Movie Script Database (IMSDb), which provides publicly available screenplays.

Why I found this dataset interesting:

I was curious to see if the language used in romantic films differs significantly from science fiction films in quantifiable ways. Movie scripts are fascinating because they contain both dialogue and scene descriptions, giving us insight into how different genres tell their stories through language.

1.2 Data Collection Process

Where I got the data: I downloaded all scripts from https://imsdb.com/ using Python web scraping with BeautifulSoup and requests.

The films I selected:

Romantic:

- The Fault in Our Stars
- The Perks of Being a Wallflower
- Pride and Prejudice
- Titanic
- 10 Things I Hate About You

Sci-Fi:

- Avatar
- The Fifth Element
- Interstellar
- Tenet
- The Martian

1.3 How I Split the Data into Documents

A major challenge was that complete film scripts are extremely long, which would give me only 10 documents total. But it was not enough for meaningful statistical analysis. So I split each script into smaller blocks based on paragraph boundaries. I filtered out very short fragments (less than 50 characters) to avoid including scene headers like "INT. BEDROOM - DAY" on their own. Each document is essentially a scene fragment, a dialogue exchange, or an action description. This approach gave me thousands of documents while preserving the local meaning within each block.

1.4 Dataset Statistics

Here's what my final dataset looks like:

Overall Dataset Statistics

Category	Number of Films	Number of Documents	Avg Documents per Film
Romantic	5	5,382	1,076
Sci-Fi	5	6,038	1,207
Total	**10**	**11,420**	**1,142**

Vocabulary statistics:

- Baseline vocabulary: 2,547 unique words
- After aggressive filtering: 1,209 unique words

1.5 Breakdown by Individual Film

Not all films contributed equally to the dataset. Here's the document count per film:

Number of Documents per Film

Film	Category	Documents
10 Things I Hate About You	Romantic	517
The Fault in Our Stars	Romantic	724
The Perks of Being a Wallflower	Romantic	939
Pride and Prejudice	Romantic	659
Titanic	Romantic	2,543
Avatar	Sci-Fi	1,874
The Fifth Element	Sci-Fi	806
Interstellar	Sci-Fi	1,536
The Martian	Sci-Fi	692
Tenet	Sci-Fi	1,130

I noticed that Titanic and Avatar are significantly larger than the others. This makes sense because both are epic, long films with detailed scripts and extensive scene descriptions. While this creates some imbalance, I decided it was acceptable because these films are important representatives of their genres.

2. Methodology

This section describes the complete analysis pipeline I implemented, including all preprocessing decisions, the rationale behind them, and the tools I used.

2.1 Text Preprocessing

I experimented with multiple preprocessing configurations to find the approach that gave the most interpretable results. Here's what I tried and why:

2.1.1 Basic Preprocessing

Lowercasing: I converted all text to lowercase to ensure that "The", "the", and "THE" were treated as the same word. This is standard practice and it's helped reduce vocabulary size without losing meaningful information.

Tokenization: I used scikit-learn's CountVectorizer with a regex pattern r'\b[a-zA-Z]{2,}\b' (later changed to {3,} in experiments) to extract only alphabetic tokens with a minimum length. This filtered out:

- Numbers and punctuation
- Single-letter tokens (which are rarely meaningful in scripts)
- Scene markers like "INT." and "EXT."

Why I chose this approach: Movie scripts contain a lot of formatting metadata (scene numbers, camera directions) that isn't relevant to genre analysis. By restricting to alphabetic tokens, I focused on the actual language used.

2.1.2 Stop Words Removal

I started with scikit-learn's built-in English stop words list, which removes common words like "the", "and", "is", etc. After running initial analyses, I noticed that character names completely dominated the results. For example, the top "romantic" words were "rose", "jack", "hazel", "gus" all character names from specific films. While technically correct (these names do appear more in romantic films), this wasn't insightful for understanding genre differences. I manually curated a list of 100+ character names from all 10 films and added them to the stop words list. This included:

- Main characters (Jake, Cooper, Rose, Elizabeth)
- Supporting characters (Quaritch, Neytiri, Darcy)
- Film titles (Titanic, Avatar)

Why this was necessary: Character names are film-specific, not genre-specific. Removing them allowed me to see the actual thematic and linguistic patterns that define each genre.

2.1.3 Document Frequency Filtering

I used two parameters to filter vocabulary based on document frequency:

min_df (minimum document frequency):

- Baseline: 5 documents
- Aggressive filtering: 10 documents
- Purpose: Remove rare words that appear in very few documents. These are often typos, unusual proper nouns, or highly specific terms that don't help with genre classification.

max_df (maximum document frequency):

- Baseline: 0.8 (80% of documents)
- Aggressive filtering: 0.7 (70% of documents)

- Purpose: Remove extremely common words that appear everywhere. Words like "looks", "goes", "says" appeared in almost every document regardless of genre, so they weren't useful for distinguishing categories.

2.2 Bag-of-Words Representation

I created document-term matrices using scikit-learn's CountVectorizer and TfidfVectorizer. Given the size of my dataset (11,420 documents), I used scipy's sparse matrix format to efficiently store the representation, since most document-word combinations have a count of zero.

For my baseline configuration, I used CountVectorizer with standard English stop words, a minimum document frequency of 5, maximum document frequency of 80%, and a minimum word length of 2 characters. This produced a vocabulary of 2,547 unique words, but character names heavily dominated the results, making it difficult to identify meaningful genre patterns.

I then tried TfidfVectorizer with the same parameters to weight words by their importance across documents. While rare but distinctive words received higher weights, the vocabulary size remained the same at 2,547 words and genre separation actually worsened, with the same topics dominating both categories.

For my final configuration, I used CountVectorizer with a custom stop words list that combined standard English stop words with over 100 manually curated character names. I also applied stricter frequency thresholds (minimum document frequency of 10, maximum of 70%) and increased the minimum word length to 3 characters. This reduced the vocabulary to 1,209 unique words but produced significantly cleaner and more interpretable results, with topics focusing on actions, settings, and themes rather than character-specific language.

2.3 Naive Bayes Probability Analysis

I used the Naive Bayes framework to identify which words are most characteristic of each genre.

2.3.1 Probability Calculation

For each word w and category c, I calculated:

$$P(w|c) = (count(w, c) + \alpha) / (total\_words(c) + \alpha \times V)$$

Where:

- count(w, c) = number of times word w appears in category c
- total_words(c) = total words in category c
- $\alpha = 1$ (add-one smoothing)
- V = vocabulary size

I used add-one smoothing because without smoothing, words that never appear in a category would have P(w|c) = 0, making log calculations undefined. Add-one smoothing ensures all probabilities are non-zero.

2.3.2 Log-Likelihood Ratio (LLR)

To identify distinctive words, I calculated:

llr(w, romantic) = log(P(w|romantic)) - log(P(w|sci-fi))

Interpretation:

- Positive values => word more associated with romantic films
- Negative values => word more associated with sci-fi films
- Magnitude => strength of association

2.4 Topic Modeling (LDA)

I used Latent Dirichlet Allocation to discover latent themes in the corpus.

Implementation: scikit-learn's LatentDirichletAllocation

I experimented with 8, 10, 12, and 15 topics. With 10 films in my corpus, 10 topics provided a good balance: After extracting topics, I manually examined the top 10-15 words in each topic and assigned descriptive labels. For each category, I computed the average topic distribution across all documents in that category, allowing me to identify which topics are most prominent in each genre.

2.5 Experimental Variations

To find the best preprocessing configuration, I systematically tested three variations:

Experiment 1: Stemming

Tool: NLTK's PorterStemmer

I applied stemming to reduce words to their root forms. Vocabulary reduced from 2,547 to 2,301 words. However, topic quality didn't improve significantly. Stemmed words like "look", "turn", "walk" still appeared frequently but weren't more interpretable. Stemming alone wasn't sufficient to solve the character name problem, so I didn't use it in my final analysis.

Experiment 2: TF-IDF Weighting

Instead of raw word counts, I weighted words by their TF-IDF scores to emphasize rare, distinctive words. Topic distributions became more balanced, but I noticed Topic 1 became dominant in BOTH categories, indicating poor genre separation. TF-IDF helped with word weighting but didn't address the fundamental issue of character names dominating the analysis.

Experiment 3: Aggressive Stopword Filtering

Combined approach:

- Custom stopword list (English + 100+ character names)
- Stricter frequency thresholds (min_df=10, max_df=0.7)
- Longer minimum word length (3 letters instead of 2)

Result:

- Vocabulary: 1,209 words
- Topics focused on actions, themes, and settings
- Clear genre separation in topic distributions
- Most interpretable results

This was my final configuration. While I lost some vocabulary, the trade-off was worth it for interpretability and meaningful genre analysis.

3. Results and Analysis

3.1 Naive Bayes: Distinctive Words by Genre

Using the log-likelihood ratio approach with aggressive filtering, I identified the words most characteristic of each genre.

3.1.1 Top Distinctive Words

Most Distinctive Words by Genre (Aggressive Filtering)

Top 10 Romantic Words

Rank	Word	Log-Likelihood Ratio
1	bedroom	8.28
2	women	7.91
3	letter	7.74
4	aunt	7.74
5	afternoon	7.69
6	dress	7.5
7	chastity	7.45
8	married	7.2
9	dance	7.15
10	tea	7.1

Top 10 Sci-Fi Words

Rank	Word	Log-Likelihood Ratio
1	probe	8.35
2	airlock	8.23
3	forest	8.28
4	banshee	8.13
5	hab	7.85
6	troopers	7.85
7	oxygen	7.6
8	planet	7.55
9	spacecraft	7.4
10	module	7.35

After removing character names, the distinctive words revealed clear thematic differences between the two genres. Romantic films emphasized domestic and intimate spaces like "bedroom," "tea," and "dance," as well as social structures through words like "women," "aunt," and "married." Communication was also prominent, particularly "letter," which reflected written correspondence as a major theme in Pride and Prejudice and other period romances. Words like "afternoon" suggested leisurely, contemplative atmospheres, while "chastity" reflected moral themes common in period romances. In contrast, sci-fi films emphasized technology and survival. Words like "probe," "airlock," "spacecraft," and "module" dominated the technical vocabulary. Survival equipment was captured through terms like "hab" (habitat from The Martian) and "oxygen." Exploration themes emerged through "forest" (Pandora in Avatar) and "planet," while military elements appeared through "troopers" and "banshee" (both from Avatar). These results made intuitive sense and demonstrated that aggressive filtering successfully captured genre-defining vocabulary rather than character-specific language.

3.1.2 Comparison: Before and After Character Name Filtering

Impact of Character Name Filtering (Top 5 Romantic Words)

Rank	Baseline (With Names)	Aggressive (Names Removed)
1	rose	bedroom
2	jack	women
3	hazel	letter
4	elizabeth	aunt
5	titanic	afternoon

Impact of Character Name Filtering (Top 5 Sci-Fi Words)

Rank	Baseline (With Names)	Aggressive (Names Removed)
1	jake	probe
2	cooper	airlock
3	protagonist	forest
4	korben	banshee
5	neytiri	hab

This comparison demonstrates why aggressive filtering was essential. The baseline approach identified film-specific proper nouns (character names, ship name) rather than genre-defining vocabulary. Only after removing these names did meaningful thematic patterns emerge.

LDA Topic Model Results - Top 10 Words per Topic (Baseline)

Topic ID	Topic Label	Word 1	Word 2	Word 3	Word 4	Word 5	Word 6	Word 7	Word 8	Word 9	Word 10
0	Pride & Prejudice Scenes	elizabeth (195.79)	hand (155.44)	opens (132.58)	way (113.87)	goes (111.12)	past (100.04)	darcy (86.14)	window (75.92)	run (73.40)	white (71.63)
1	Tenet - Action & Time	protagonist (374.26)	kat (223.91)	away (204.67)	eyes (202.71)	turns (170.95)	just (163.84)	watches (163.58)	sator (142.86)	look (140.50)	black (138.23)
2	Avatar - Pandora Life	room (258.08)	water (212.58)	sits (155.91)	walks (133.84)	neytiri (110.78)	jake (107.32)	suddenly (97.20)	gets (92.00)	mr (88.36)	puts (82.60)
3	Fifth Element & Charlie	sees (178.68)	charlie (176.24)	int (150.54)	house (84.65)	korben (84.42)	leeloo (78.65)	high (74.49)	air (68.01)	enters (66.63)	young (62.68)
4	Interstellar - Space Mission	looks (432.90)	door (276.23)	cooper (161.85)	open (154.87)	looking (146.47)	jake (137.89)	case (132.25)	begins (121.71)	man (117.67)	steps (113.96)
5	Mixed Romantic Moments	looks (164.51)	brand (149.85)	patrick (109.60)	night (109.17)	time (106.06)	cooper (103.16)	close (93.04)	sam (87.09)	little (82.95)	floor (75.11)
6	General Action Scenes	starts (112.58)	wall (102.49)	holds (99.99)	old (83.74)	tries (79.18)	trying (77.95)	gun (76.18)	standing (75.05)	glass (73.02)	day (71.85)
7	Titanic & Emotional Scenes	rose (320.57)	jack (184.87)	face (164.33)	hands (163.02)	head (160.96)	hazel (143.95)	takes (143.82)	gus (138.27)	pulls (132.46)	moment (124.81)
8	The Martian & Avatar Mix	mark (224.42)	jake (161.86)	moves (109.74)	light (105.50)	feet (103.52)	table (91.35)	quaritch (78.19)	screen (75.04)	ground (74.16)	slowly (64.11)
9	Titanic - Ship & Sea	like (355.12)	ship (187.23)	deck (140.58)	grabs (134.35)	inside (126.88)	cooper (112.82)	boat (103.46)	comes (99.82)	checks (82.78)	long (82.11)

This table presents the top 10 words for each of the 10 topics discovered by LDA, along with their probability scores. The topics reveal a mix of film-specific and thematic patterns. Some topics are clearly dominated by a single film: Topic 7 (Titanic & Emotional Scenes) features "rose" and "jack" as its top words, while Topic 1 (Tenet - Action & Time) is led by "protagonist" and "kat." Other topics capture broader themes that span multiple films, such as Topic 6 (General Action Scenes), which contains generic action words like "starts," "wall," "holds," and "gun" without being tied to any specific film. Notably, character names like "cooper," "jake," and "brand" appear across multiple topics, indicating that these characters play significant roles in several different scene types. Topic 9 (Titanic - Ship & Sea) is particularly interesting as "ship," "deck," and "boat" suggest maritime settings that could apply to both Titanic and sci-fi spacecraft scenes.

3.2.1 Topic Modeling

To identify latent themes in the corpus, I ran Latent Dirichlet Allocation using scikit-learn's LatentDirichletAllocation with 10 topics. I experimented with 8, 10, 12, and 15 topics before settling on 10, as it produced the most interpretable results without being too general or too redundant. I used the aggressive filtering configuration for the final model since it gave the cleanest vocabulary. After fitting the model, I manually examined the top 15 words in each topic and assigned descriptive labels based on the dominant themes I observed. The resulting topics captured action-oriented and scene-based patterns rather than film-specific vocabulary, which made sense given that scripts from both genres share common narrative structures like movement, dialogue, and setting descriptions.

Average Topic Distribution by Category

Topic	Label	Romantic	Sci-Fi
Topic 0	Pride & Prejudice Scenes	0.0997	0.0829
Topic 1	Tenet - Action & Time	0.1054	0.1185
Topic 2	Avatar - Pandora Life	0.0956	0.0917
Topic 3	Fifth Element & Charlie	0.0992	0.0847
Topic 4	Interstellar - Space Mission	0.0910	0.1257
Topic 5	Mixed Romantic Moments	0.1090	0.0984
Topic 6	General Action Scenes	0.1000	0.0894
Topic 7	Titanic & Emotional Scenes	0.1198	0.0841
Topic 8	The Martian & Avatar Mix	0.0759	0.1189
Topic 9	Titanic - Ship & Sea	0.1043	0.1057

The topic distribution showed clear differences between genres. Romantic films had the highest association with Topic 7 (Titanic & Emotional Scenes) at 11.98% and Topic 5 (Mixed Romantic Moments) at 10.90%, reflecting their focus on emotional relationships. Sci-fi films showed stronger association with Topic 4 (Interstellar - Space Mission) at 12.57% and Topic 8 (The Martian & Avatar Mix) at 11.89%, emphasizing space exploration and technology. Interestingly, some topics appeared in both genres' top five. Topic 1 (Tenet - Action & Time) ranked third for both romantic and sci-fi films, and Topic 9 (Titanic - Ship & Sea) appeared in both top fives. This overlap suggests that action sequences and ship settings work across both genres. The balanced distribution, with most topics between 8-13%, shows that the 10-topic model captured diverse themes without any single topic dominating.

Top 5 Topics per Category

ROMANTIC - Top 5 Topics

Rank	Topic	Label	Avg. Probability
1	Topic 7	Titanic & Emotional Scenes	0.1198
2	Topic 5	Mixed Romantic Moments	0.1090
3	Topic 1	Tenet - Action & Time	0.1054
4	Topic 9	Titanic - Ship & Sea	0.1043
5	Topic 6	General Action Scenes	0.1000

SCI-FI - Top 5 Topics

Rank	Topic	Label	Avg. Probability
1	Topic 4	Interstellar - Space Mission	0.1257
2	Topic 8	The Martian & Avatar Mix	0.1189
3	Topic 1	Tenet - Action & Time	0.1185
4	Topic 9	Titanic - Ship & Sea	0.1057
5	Topic 5	Mixed Romantic Moments	0.0984

3.3 Experimental Comparisons

I tested three preprocessing variations. Here's how they compared:

Preprocessing Configuration Comparison

Configuration	Vocabulary Size	Top Romantic Word	Top Sci-Fi Word	Topic Quality	Genre Separation
Baseline	2547	rose	jake	Poor (names)	Good
Stemming	2301	charli	jake	Poor (names)	Moderate
TF-IDF	2547	protagonist	protagonist	Moderate	Poor
Aggressive Filtering	**1209**	**bedroom**	**probe**	**Excellent**	**Good**

The three experimental configurations revealed important differences in preprocessing effectiveness. Stemming reduced vocabulary by 10% but failed to solve the character name problem,

as names remained dominant in the results. TF-IDF weighting caused poor genre separation, with Topic 1 dominating both categories equally, indicating that the weighting scheme obscured genre-specific patterns. Aggressive filtering achieved the best results overall, producing the most interpretable vocabulary with clear genre separation and topics that focused on thematic content rather than character names. However, this approach involved a significant trade-off: aggressive filtering removed 52% of the vocabulary. Despite this reduction, the remaining words proved far more meaningful for genre analysis, demonstrating that vocabulary size alone does not guarantee interpretability.

Top 3 Topics per Category - Configuration Comparison

Configuration	Romantic Top 3	Sci-Fi Top 3
Baseline	T7 (Titanic & Emotional Scenes): 0.120 T5 (Mixed Romantic Moments): 0.109 T1 (Tenet - Action & Time): 0.105	T4 (Interstellar - Space Mission): 0.126 T8 (The Martian & Avatar Mix): 0.119 T1 (Tenet - Action & Time): 0.119
Stemming	T0 (Pride & Prejudice Scenes): 0.137 T8 (The Martian & Avatar Mix): 0.120 T4 (Interstellar - Space Mission): 0.108	T9 (Titanic - Ship & Sea): 0.124 T2 (Avatar - Pandora Life): 0.113 T4 (Interstellar - Space Mission): 0.106
TF-IDF	T1 (Tenet - Action & Time): 0.123 T9 (Titanic - Ship & Sea): 0.111 T6 (General Action Scenes): 0.111	T1 (Tenet - Action & Time): 0.122 T7 (Titanic & Emotional Scenes): 0.111 T5 (Mixed Romantic Moments): 0.106
Aggressive	T4 (Interstellar - Space Mission): 0.115 T7 (Titanic & Emotional Scenes): 0.111 T2 (Avatar - Pandora Life): 0.103	T9 (Titanic - Ship & Sea): 0.117 T7 (Titanic & Emotional Scenes): 0.114 T6 (General Action Scenes): 0.113

Comparing the four preprocessing configurations revealed how different approaches affected topic distributions and genre separation. The baseline configuration showed clear genre distinction, with romantic and sci-fi films having completely different top-3 topics. Stemming maintained reasonable separation but changed which topics dominated each category. TF-IDF performed poorly for genre separation, as Topic 1 dominated both romantic and sci-fi films equally, indicating that the weighting scheme obscured genre-specific patterns. Aggressive filtering produced the cleanest results with good genre separation, though the distributions were more balanced across topics compared to the baseline. Notably, Topic 7 appeared in the top-3 for both genres under aggressive filtering, suggesting it captured universal storytelling elements. Overall, while baseline and aggressive filtering both achieved good genre separation, aggressive filtering's advantage lay in its interpretable vocabulary rather than sharper distributional differences.

4. Discussion

4.1 Insights from the Dataset

My analysis of 10 movie scripts revealed clear linguistic differences between romantic and sci-fi films. Romantic films emphasized intimate spaces and relationships, with words like "bedroom," "letter," "aunt," and "married" dominating the distinctive vocabulary. These films focused on personal connections and domestic settings. In contrast, sci-fi films centered on technology and survival, with "probe," "airlock," "spacecraft," and "oxygen" emerging as the most distinctive terms. Even emotional moments in sci-fi were framed through technological language.

Interestingly, both genres shared similar storytelling mechanics but applied them to different contexts. Romantic characters watched each other, while sci-fi characters monitored equipment. The biggest challenge I encountered was character names dominating the initial results until I manually removed them. This revealed an important insight: statistically correct results aren't always meaningful or insightful. The filtering process dramatically improved interpretability,

transforming the analysis from a list of character names into a true comparison of genre-defining vocabulary.

4.2 Personal Lessons Learned

This project taught me several important lessons about NLP in practice. Preprocessing had a much larger impact than I expected: stemming didn't help, TF-IDF worsened results, but removing character names transformed the analysis, showing there's no universal best approach. Real-world data proved messier than textbook examples, with movie scripts requiring extensive cleaning due to formatting quirks and scene directions. I spent more time on data preparation than analysis itself. Topic modeling revealed patterns rather than stories, discovering action types instead of film-specific topics, which unexpectedly showed that storytelling structures recur across different films. Manual inspection proved essential, as automated metrics missed the character name problem that I only caught by examining outputs directly. Every preprocessing choice involves trade-offs: aggressive filtering removed 52% of vocabulary but dramatically improved interpretability. If I repeated this project, I'd use Named Entity Recognition to detect names automatically, experiment with bigrams to capture phrases, and test additional genres to see if these patterns generalize.