

셰익스피어 극 작품 장르별 주제, 감정 분석 및 크리스토퍼 말로우 작품과의 비교분석

1. 서론

가. 분석 목적

셰익스피어와 크리스토퍼 말로우는 모두 르네상스 시대에 활동한 대표적 극작가들이다. 당 시에는 정치적, 사회적 변동이 컸고, 종교적 갈등 역시 존재한 시대이다. 문학은 시대를 반영하기에 두 작가의 작품들에는 야망, 도덕적 딜레마 등의 소재가 주로 사용되었다. 특히 셰익스피어는 해당 주제뿐만 아니라 여러 장르의 극 작품들을 탄생시켰다. 셰익스피어와 크리스토퍼 말로우는 각 극단에서 뛰어난 작품들을 내세우며, 경쟁 관계가 되었으며, 크리스토퍼 말로우의 작품은 셰익스피어의 작품에 크고 작은 영향을 주었으며, 문제 등에서 유사성을 확인할 수 있다. 이에 본 연구는 르네상스 영문학의 대표적 인물인 셰익스피어의 극 작품의 감정 분석을 통해 그 서사 구조를 분석하고 각 장르별 주제의 차이를 확인할 수 있는지 분석한다. 또한, 셰익스피어의 작품들 중 경쟁 관계이자 영향을 받은 크리스토퍼 말로우의 작품들과의 감정분석 그래프 비교와 문장 유사성 확인을 통해 그 유사성이 존재하는지 확인하는 것이 목적이다.

크리스토퍼 말로우의 작품은 'The Jew of Malt(유대인의 비극)', Tamburlaine the Great (탐벌레인 대왕)', 'The Tragic History of Doctor Faustus(파우스트 박사의 비극)'를 선정 하였고 쌍을 이루는 셰익스피어 작품은 각각 'The Merchant of Venice(베니스의 상인)', 'Macbeth(맥베스)', 'Richard III(리처드 3세)'로 선정하였다.

나. 데이터 개요

1) 데이터 출처:

a. 셰익스피어 작품: Kaggle(csv)

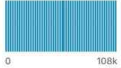



<https://www.kaggle.com/datasets/guslovesmath/shakespeare-plays-dataset>

b. 크리스토퍼 말로우 작품: Project Gutenberg(html)

https://www.gutenberg.org/ebooks/search/?query=Christopher+Marlowe&submit_search=Go%21

2) 데이터 유형:

a. 셰익스피어 데이터

| About this file Add Suggestion | | | | | | | | | |
|--|--|--|--|---|---|---|--|----------------|--|
| File: shakespeare_plays.csv | | | | | | | | | |
| This CSV file contains 108,093 lines of dialogue from all the plays written by William Shakespeare, meticulously organized and detailed for easy analysis. Each entry in the dataset is marked with metadata about the play name, genre, character name, act, scene, and sentence number, alongside the text of the dialogue and the gender of the character delivering it. The dataset is designed to facilitate both simple queries and complex linguistic analyses, suitable for educators, students, researchers, and enthusiasts interested in exploring the works of Shakespeare in depth. | | | | | | | | | |
| # Index | Δ play_name Play Name | Δ genre Genres | Δ character Characters | # act Acts | # scene Scenes | # sentence Sentences | Δ text Text | Δ sex Sex | |
|  | Hamlet 4% Coriolanus 3% Other (100309) 93% | Comedy 42% Tragedy 29% Other (30719) 28% | Gloucester 2% Falstaff 2% Other (104505) 97% |  |  |  | 106601 unique values | male female | |
| 0 | All's Well That Ends Well | Comedy | Countess | 1 | 1 | 1 | In delivering my son from me, I bury a second husband. | female | |
| 1 | All's Well That Ends Well | Comedy | Bertram | 1 | 1 | 2 | And I in going, madam, weep o'er my father's death | male | |
| 2 | All's Well That Ends Well | Comedy | Bertram | 1 | 1 | 3 | anew: but I must attend his majesty's command, to | male | |
| 3 | All's Well That Ends Well | Comedy | Bertram | 1 | 1 | 4 | whom I am now in ward, evermore in subjection. | male | |
| 4 | All's Well That Ends Well | Comedy | Lafeu | 1 | 1 | 5 | You shall find of the king a husband, madam; you, | male | |
| 5 | All's Well That Ends Well | Comedy | Lafeu | 1 | 1 | 6 | sir, a father: he that so generally is at all times | male | |

칼럼의 왼쪽부터 제목(play_name), 장르(genre), 등장인물(character), 극(act), 장면(scene), 대사 라인 넘버(sentence), 대사(text), 성별(sex) 이루어진 원본 데이터

b. 크리스토퍼 말로우 작품(3개)

```
<h2>
  THE JEW OF MALTA.
</h2>
<pre>
  Enter MACHIAVEL.

MACHIAVEL. Albeit the world think Machiavel is dead,
Yet was his soul but flown beyond the Alps;
And, now the Guise <a href="#linknote-11" id="linknoteref-11" class="pginternal">11</a> is dead, is
come from France,
To view this land, and frolic with his friends.
To some perhaps my name is odious;
But such as love me, guard me from their tongues,
And let them know that I am Machiavel,
And weigh not men, and therefore not men's words.
Admir'd I am of those that hate me most:
Though some speak openly against my books,
Yet will they read me, and thereby attain
To Peter's chair; and, when they cast me off,
Are poison'd by my climbing followers.
I count religion but a childish toy,
And hold there is no sin but ignorance.
Birds of the air will tell of murders past!
I am asham'd to hear such fooleries.
Many will talk of title to a crown:
What right had Caesar to the empery? <a href="#linknote-12" id="linknoteref-12" class="pginternal">12</a>
```

위 사진은 전처리 전 html 파일 원본

| play_name | character | act | scene | sentence | text |
|------------------|-----------|--------|-------|----------|--|
| The Jew of Malta | BARABAS | ACT I. | none | 1 | So that of thus much that return was made; |
| The Jew of Malta | BARABAS | ACT I. | none | 2 | And of the third part of the Persian ships |
| The Jew of Malta | BARABAS | ACT I. | none | 3 | There was the venture summ'd and satisfied. |
| The Jew of Malta | BARABAS | ACT I. | none | 4 | As for those Samnites, and the men of Uz, |
| The Jew of Malta | BARABAS | ACT I. | none | 5 | That bought my Spanish oils and wines of Greece, |
| The Jew of Malta | BARABAS | ACT I. | none | 6 | Here have I purs'd their paltry silverlings. |
| The Jew of Malta | BARABAS | ACT I. | none | 7 | Fie, what a trouble 'tis to count this trash! |
| The Jew of Malta | BARABAS | ACT I. | none | 8 | Well fare the Arabians, who so richly pay |

전처리 후 csv 파일로 저장한 데이터

3) 분석 대상:

셰익스피어 극 작품 대사들의 감정 분석 점수를 구해, 장르별로 감정 변화 양상이 어떻게 달라지는지, 같은 장르에서는 유사성이 확인되는지 분석한다.

그 후 크리스토퍼 말로우의 극 작품 역시 동일하게 감정 분석 점수를 구해 셰익스피어 작품 3개와 비교분석을 진행한다.

4) 데이터 요약:

셰익스피어 작품 데이터의 경우 크리스토퍼 말로우 작품과는 다르게 장르에 대한 구분을 통해 감정 분석을 비교하기 위해 해당 칼럼이 필요하다. 다만, 크리스토퍼 말로우의 작품 경우 비교 대상을 미리 선정해 진행하기에 대사과 그에 대한 감정분석, 극과 장면의 구분만 필요하기에 그에 맞게 전처리를 진행하였다. 전체적으로 두 데이터 모두 두 작가의 작품을 등장 인물의 대사, 대사의 라인 별로 구분해 구성된 데이터셋이다.

다. 작품 쌍 선정 이유

1) "유대인의비극" vs "베니스의 상인"

두 작품 모두 유대인 인물을 중심으로 사회와의 갈등을 다루며, 캐릭터의 역할과 갈등 구조를 통해 당대 사회의 편견과 윤리적 문제를 주제로 한다.

2) "템벌레인 대왕" vs 셰익스피어의 "맥베스"

두 주인공이 권력을 위해 어떤 과정을 거치는지와 그 과정에서 변하는 심리를 다룬 작품이며, 왕권을 차지하려는 야망과 그로 인한 비극적인 결말의 공통점을 가진다.

3) "파우스트스 박사의 비극" vs "리처드 3세"

두 작품 모두 인간의 야망과 도덕적 딜레마를 다루며 주인공이 파멸에 이르는 과정의 서사를 가진다.

2. 데이터 전처리(정확한 분석을 위해 필요한 전처리들/1page~3page)

가. html 파일 필요값 추출 및 csv파일 저장

1) 데이터 처리 방식

불필요 태그 및 무대 지시어 삭제

```
import re

# Load the file
with open(input_file, 'r', encoding='utf-8') as file:
    content = file.read()

# Remove all <a>...</a> tags, including those with attributes
content = re.sub(r'<a\b[^\>]*.*?</a>', '', content)

# Remove all stage directions within square brackets []
content = re.sub(r'\[.*?\]', '', content)

# Save the cleaned content
with open(output_file, 'w', encoding='utf-8') as file:
    file.write(content)

print("All <a>...</a> tags (including with attributes), stage directions in [], and indentation have been removed.")

from bs4 import BeautifulSoup

# HTML 파일 불러오기
with open(input_file, 'r', encoding='utf-8') as file:
    soup = BeautifulSoup(file, 'html.parser')

# <pre> 태그에서 들여쓰기가 두 번 된 무대 지시어 제거
for tag in soup.find_all('pre'):
    lines = tag.get_text().splitlines()
    new_lines = []

    for line in lines:
        if not line.startswith("      "): # 들여쓰기가 두 번 된 줄만 제거
            new_lines.append(line)
        else:
            new_lines.append("") # 빈 줄 추가

    # 수정된 내용으로 <pre> 태그에 다시 설정
    tag.string = "\n".join(new_lines)

# 수정된 파일 저장
with open(output_file, 'w', encoding='utf-8') as file:
    file.write(str(soup))
```

위 코드는 html 파일에서 불필요한 정보를 미리 제거하기 위한 코드

이름 태그 구분

```
from bs4 import BeautifulSoup
import re

# HTML 파일 불러오기
with open(input_file, 'r', encoding='utf-8') as file:
    soup = BeautifulSoup(file, 'html.parser')

# <pre> 태그 내 등장인물 이름에 <name> 태그 추가
for tag in soup.find_all('pre'):
    lines = tag.get_text().splitlines()
    new_lines = []

    for line in lines:
        # '등장인물.대사' 형식 찾기
        match = re.match(r"^\s*([A-Z]+(?:[\s-][A-Z]+)*)\.\s*(.*)", line)
        if match:
            # 등장인물 이름과 대사를 각각 <name> 태그와 함께 추가
            character = match.group(1)
            dialogue = match.group(2).strip()
            new_lines.append(f"    <name>{character}<name>\n        {dialogue}")
        else:
            new_lines.append(line) # 나머지 줄은 그대로 추가

# 수정된 내용으로 <pre> 태그 업데이트
tag.string = "\n".join(new_lines)

# 수정된 파일 저장
with open(output_file, 'w', encoding='utf-8') as file:
    file.write(str(soup))
```

위 코드는 html 파일에서 등장인물과 대사의 구분이 ‘.’표시로만 되어 있어
데이터 처리를 위해 등장인물과 대사를 구분하는 태그를 추가한 것이다.

문장 끝 -부분 제거

```
from bs4 import BeautifulSoup

def remove_hyphens_in_pre_tags(file_path):
    # HTML 파일 읽기
    with open(file_path, 'r', encoding='utf-8') as file:
        html_content = file.read()

    # BeautifulSoup로 HTML 파싱
    soup = BeautifulSoup(html_content, 'html.parser')

    # 모든 <pre> 태그 찾기
    for pre in soup.find_all('pre'):
        # <pre> 태그 내 텍스트 줄 단위로 처리
        lines = pre.get_text().splitlines()
        cleaned_lines = [line.rstrip('--') if line.endswith(('-', '-')) else line for line in lines]

        # <pre> 태그 내 텍스트 업데이트
        pre.string = "\n".join(cleaned_lines)

    # 수정된 HTML을 파일에 저장
    with open(file_path, 'w', encoding='utf-8') as file:
        file.write(str(soup))

# 사용 예시
remove_hyphens_in_pre_tags("C:\\Users\\kang8\\OneDrive\\바탕 화면\\빅데이터융합개론\\크리스토퍼 말로우\\파우스트 박사\\pg779-images.html")
```

데이터 처리 중 문장 끝에 ‘-’ 기호가 존재할 시 csv 파일에 저장이 제대로 되지 않아

그 값을 제거한 후 셰익스피어 작품의 양식과 동일하게 필요 정보를 csv파일로 저장 하였다.

나. 결측치 처리

1) 결측치 종류:

- a. ACT 또는 SCENE이 존재하지 않는 크리스토퍼 말로우 작품이 존재
- b. 등장인물의 대사 사이 줄바꿈 공백이 존재하는 값 존재

2) 처리 방법:

- a. 'none'으로 처리
- b. 줄바꿈 삭제

3) 결측치 예 및 처리 방식:

a.

| act | scene |
|-----|-------|
| 1 | none |
| 1 | none |
| 1 | none |
| 1 | none |
| 1 | none |
| 1 | none |
| 1 | none |
| 1 | none |
| 1 | none |

b. 처리 방식 코드

```
import pandas as pd

# Load the CSV file
file_path = "C:\\Users\\kang8\\OneDrive\\바탕 화면\\빅데이터융합개론\\크리스토퍼 말로우\\파우스트 박샤\\out_put.csv"
df = pd.read_csv(file_path, encoding='utf-8')

# Remove rows where 'text' contains only whitespace or is empty
df['text'] = df['text'].replace(r'^\s*$', pd.NA, regex=True)
df = df.dropna(subset=['text']).reset_index(drop=True)

# Reassign sentence numbers, starting from 1 each time a character appears with a new line
df['sentence'] = df.groupby((df['character'] != df['character'].shift()).cumsum()).cumcount() + 1

# Overwrite the original file with the modified DataFrame
df.to_csv(file_path, index=False)

print("The file has been updated successfully.")
```

다. 데이터 스케일링

1) 감정 분석 점수값 소수점 반올림

감정 분석의 점수가 필요 이상으로 세세하게 측정되어 소수점 4자리에서 반올림해 감정 점수를 측정하였다.

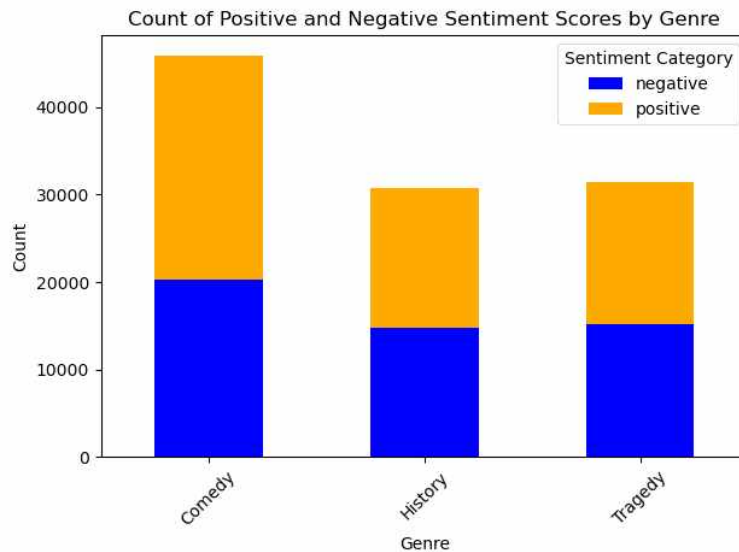
2) 감정 변화 그래프 정규화 및 스무딩

감정 점수 변화 양상 그래프를 시각화할 시, 각 작품의 길이가 다르기 때문에 x축 값을 0~1의 값으로 정규화를 한 후 비교하였고, y축 값은 스무딩을 통해 그래프를 시각화함.

3. 데이터 분석

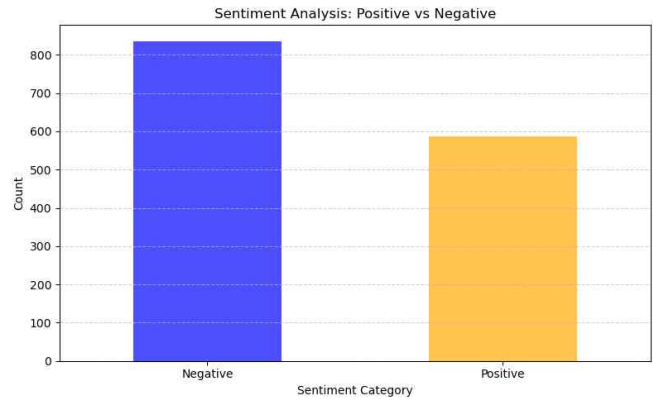
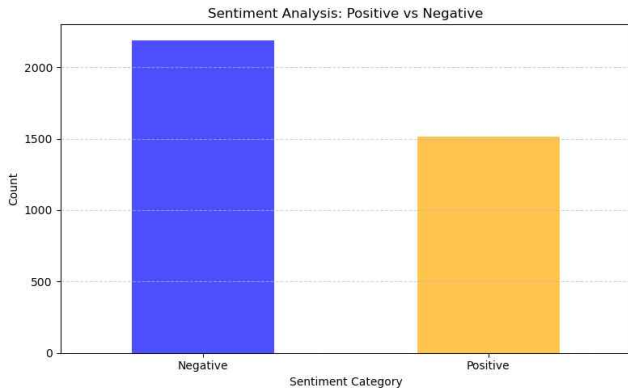
가. 기술 통계 분석

1) 셰익스피어 장르별 감정 통계



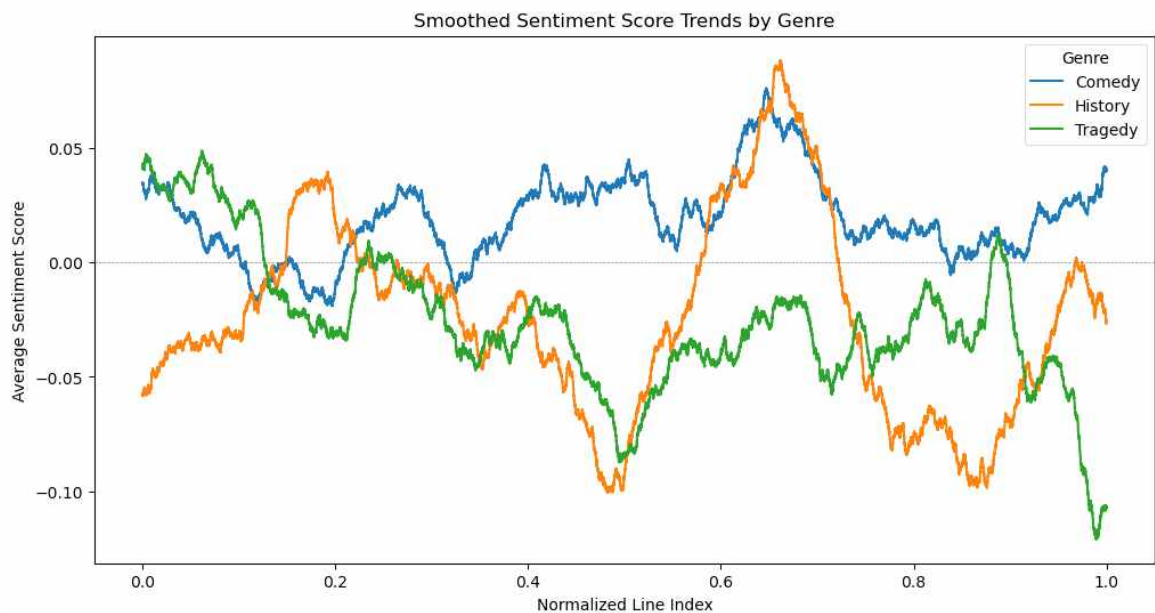
각각 감정분석 결과를 통해 점수가 0보다 크면 Positive(긍정적 감정), 0보다 작으면 Negative(부정적 감정)으로 처리해 장르별 감정 통계를 나타낸 차트이다. 물론 각 장르별 작품의 수, 대사의 양이 작품마다 다르지만, 희극, 비극, 역사극 모두 부정적 대사보다는 긍정적 대사가 많은 것을 확인할 수 있다. 다만, 본 연구의 목적은 감정분석을 통한 서사구조 확인 및 비교 분석이기에 총계적 수치는 무의미하지만, 비극에서도 긍정적 대사가 훨씬 많은 것을 확인할 수 있다.

2) 셰익스피어와 크리스토퍼 말로우의 감정 통계치



위 두 차트는 각각 셰익스피어의 '리처드 3세'(원) 크리스토퍼 말로우의 '파우스트 박사의 비극'(오)의 긍정과 부정 통계치이다. 두 작품 모두 부정의 감정이 많으며, 이는 비교군인 다른 모든 작품에서 동일하게 나타나는 통계 구조이기에 나머지는 생략한다.

나. 상관관계 분석



1) 셰익스피어 극 작품 장르별 감정 변화 양상

위 그래프에서는 셰익스피어 작품들 중 희극(Comedy), 비극(Tragedy), 역사극(History)의 감정 변화 양상을 나타낸 그래프이다. 비교분석을 통해 알 수 있는 점은 희극과 비극이 전반부(0.0~0.4)에서는 비슷한 감정 변화 양상이 나타난다는 것이다. 그러다 중반부(0.5 부근) 이후부터 희극은 긍정적 감정이 많아지지만, 비극은 부정적 감정으로 이야기의 서사가 진행되는다는 것을 알 수 있다. 후반부(0.6~1.0)부터 감정의 상/하향 구조가 비슷해지면서 결말

부분으로 이야기가 마무리될 때 희극은 긍정의 감정으로 비극은 부정의 감정으로 치달는 모습을 확인할 수 있다.

역사극의 경우에는 감정의 변화 폭이 가장 큰 것을 확인할 수 있는데, 희극과 비극을 섞어 놓은 듯한 변화 양상을 보인다. 물론 역사극의 초반부는 희/비극과 완전히 다른 양상을 보이지만 0.2~0.5+ 부근에서는 비극과 비슷한 구조이며, 그 이후로는 희극의 구조와 유사하다는 것이 특징이다.

결론적으로 세 가지 장르 모두 전반부, 중반부, 후반부로 이야기가 나뉠 때 나타나는 감정 폭의 변화하는 양상은 어느 정도 일치하는 경향을 보인다. 이는 셰익스피어는 장르가 다른 작품이더라도 감정의 폭 변화 시점은 모두 비슷하다는 결론을 낼 수 있다.

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# 데이터 로드
df = pd.read_csv("C:\\Users\\kang8\\OneDrive\\바탕 화면\\빅데이터융합개론\\셰익스피어\\archive\\archive\\RoBERTA_output_with_sentiment_scores_round.csv")

# 'genre' 컬럼에서 오타자 수정 ('Tragity' -> 'Tragedy')
df['genre'] = df['genre'].replace('Tragity', 'Tragedy')

# 감정 점수 컬럼이 누락된 경우 NaN 제거
df = df.dropna(subset=['sentiment_score'])

# 각 장르별로 라인 번호를 재정의하여 x값이 0에서 시작하도록 정규화
df['line_index'] = df.groupby('genre').cumcount()

# 각 장르별 최대 line_index를 계산하여 정규화
max_line_index = df.groupby('genre')['line_index'].transform('max')
df['normalized_line_index'] = df['line_index'] / max_line_index

# 장르별 평균 감정 점수 변화 계산
genre_sentiment_trends = df.groupby(['genre', 'normalized_line_index'])['sentiment_score'].mean().unstack(level=0)

# 이동 평균(스무딩)을 위한 함수 정의
def moving_average(data, window_size=10000):
    return data.rolling(window=window_size, min_periods=1, center=True).mean()

# 그래프 생성
plt.figure(figsize=(12, 6))

# 각 장르별 감정 점수에 대해 스무딩 처리 후 그래프 출력
for genre in genre_sentiment_trends.columns:
    smoothed_sentiment = moving_average(genre_sentiment_trends[genre])
    plt.plot(genre_sentiment_trends.index, smoothed_sentiment, label=genre)

# y=0에 점선 기준선 추가
plt.axhline(0, color='gray', linestyle='--', lw=0.5) # 점선으로 기준선 추가

# 그래프 스타일 설정
plt.title('Smoothed Sentiment Score Trends by Genre')
plt.xlabel('Normalized Line Index')
plt.ylabel('Average Sentiment Score')
plt.legend(title='Genre')
plt.show()
```

해당 작업을 위한 코드

2) 셰익스피어 작품 장르별 주제어 분석

```
Comedy - Perplexity: -9.899027737875793
Comedy - Coherence Score: 0.3093564756400327
Comedy - Top 5 Topics:
Topic 0: 0.012*let + 0.009*o, + 0.007*fair + 0.007*that's + 0.007*shall + 0.007*good + 0.005*him, + 0.005*i'll + 0.005*lord + 0.004*life
Topic 1: 0.022*shall + 0.016*know + 0.012*sir, + 0.012*good + 0.011*think + 0.010*say + 0.009*love + 0.008*no, + 0.008*'tis + 0.008*you,
Topic 2: 0.012*tell + 0.010*you, + 0.009*you. + 0.009*i'll + 0.009*speak + 0.008*thee, + 0.008*pray + 0.007*it, + 0.006*me, + 0.006*master
Topic 3: 0.017*come + 0.011*o + 0.009*ay, + 0.008*sir + 0.008*now, + 0.006*till + 0.005*good + 0.005*what, + 0.005*great + 0.005*let
Topic 4: 0.019*like + 0.012*why, + 0.011*come, + 0.008*shall + 0.008*him, + 0.007*doth + 0.007*make + 0.005*better + 0.005*bear + 0.005*sweet

=====

History - Perplexity: -9.777251869470444
History - Coherence Score: 0.407917055049783
History - Top 5 Topics:
Topic 0: 0.012*good + 0.010*duke + 0.009*lord + 0.007*hear + 0.006*bear + 0.006*heart + 0.005*bid + 0.005*king, + 0.004*lord; + 0.004*gentle
Topic 1: 0.007*poor + 0.007*man + 0.006*no, + 0.006*royal + 0.006*heaven + 0.005*say, + 0.005*brother + 0.005*make + 0.005*bloody + 0.005*like
Topic 2: 0.009*love + 0.008*now, + 0.007*fair + 0.007*let + 0.007*king + 0.006*soul + 0.006*noble + 0.006*blood + 0.005*it. + 0.005*day
Topic 3: 0.037*shall + 0.010*me, + 0.010*'tis + 0.010*tell + 0.009*good + 0.009*you, + 0.007*sir + 0.006*god + 0.005*well, + 0.005*him.
Topic 4: 0.014*lord, + 0.013*let + 0.010*come + 0.009*did + 0.008*o + 0.008*richard + 0.007*o, + 0.007*then, + 0.007*king + 0.006*me.

=====

Tragedy - Perplexity: -9.803674087546066
Tragedy - Coherence Score: 0.43781200893380107
Tragedy - Top 5 Topics:
Topic 0: 0.034*shall + 0.009*come + 0.008*lord + 0.006*no, + 0.006*comes + 0.004*he's + 0.004*tears + 0.004*nature + 0.004*emperor + 0.003*say
Topic 1: 0.010*did + 0.010*man + 0.009*why, + 0.009*speak + 0.008*ay, + 0.007*great + 0.007*what, + 0.007*make + 0.006*like + 0.006*thee,
Topic 2: 0.015*o + 0.015*know + 0.014*o, + 0.014*you, + 0.013*i'll + 0.009*me, + 0.008*let + 0.007*say + 0.007*look + 0.006*hear
Topic 3: 0.017*let + 0.013*good + 0.011*come, + 0.008*now, + 0.008*sweet + 0.006*time + 0.006*men + 0.005*old + 0.005*then, + 0.005*well,
Topic 4: 0.011*love + 0.009*doth + 0.009*sir, + 0.007*poor + 0.007*till + 0.007*you, + 0.005*'tis + 0.005*bear + 0.004*set + 0.004*i'll
```

위는 LDA 모델을 활용해 각 장르별 상위 5개의 주제어를 선택해 출력한 결과이다. 각 장르별로 분석 결과를 정리해볼 수 있다. (각 장르별 Perplexity와 Coherence는 성능지표.)

a. 희극

- **주요 주제:** 주로 경쾌한 대사들이 나타나며, “good“과 ”love“ 등의 단어가 주제어로 풀려진 것을 볼 수 있다.

b. 역사극

- **주요 주제:** 'lord'와 'king' 같은 단어들이 주제어로 선택돼 작품의 속 귀족적 배경, 권력 구조 등을 확인할 수 있다. 또한 'blood', 'royal', 'noble'등의 단어들로 권력, 충성심 등을 강조하고 있다.

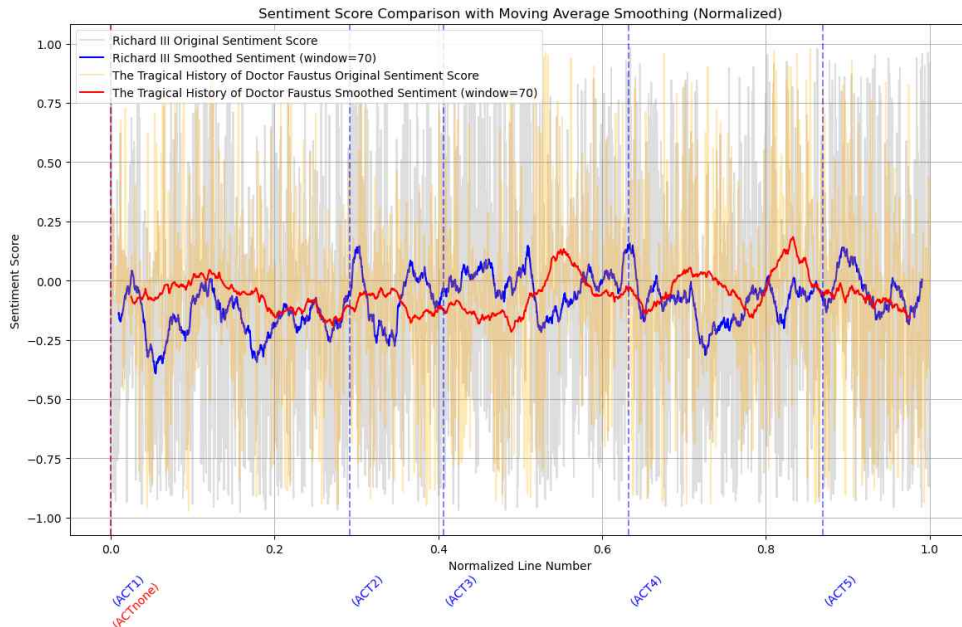
c. 비극

- **주요 주제:** 'tear'와 같은 단어가 선택되 비극의 슬픈 감정을 드러냄.

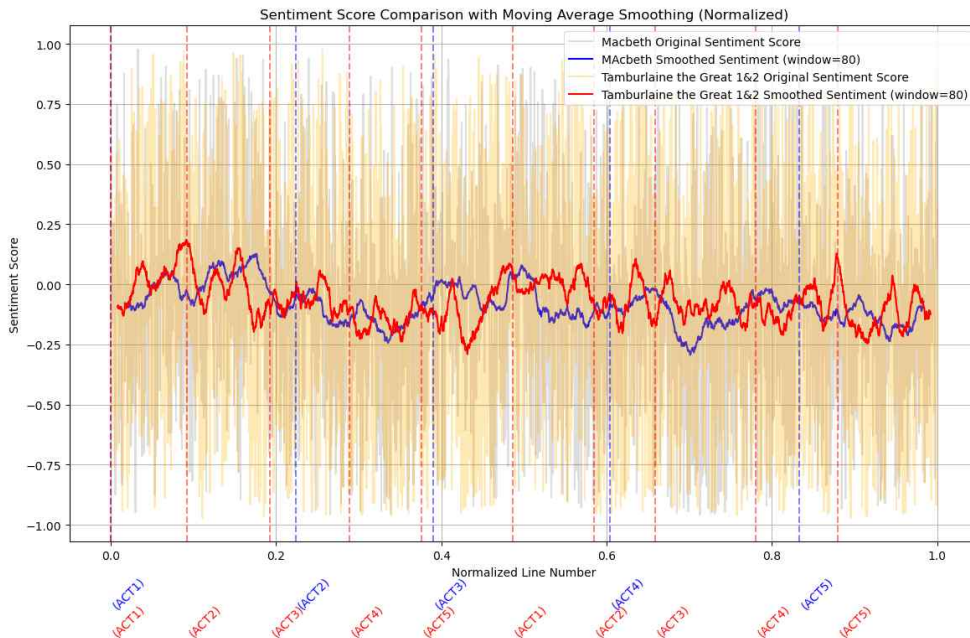
주제어 분석 결과에 대한 총 분석 시, 해당 모델을 르네상스 극 작품에 적용하기엔 무리가 있으며, 의미는 있으나 직접적인 작품의 주제를 내포하는 단어를 많이 선택하지는 못했다. 다만 극 별로 'love' / 'tear' 차이가 명확한 단어들이 각각 희/비극의 주제어로 선택되어 어느정도 참고할 만한 분석 결과를 내주었다.

3) 셰익스피어와 크리스토퍼 말로우 작품의 감정 변화 양상

a. 리처드 3세 vs 파우스트 박사의 비극



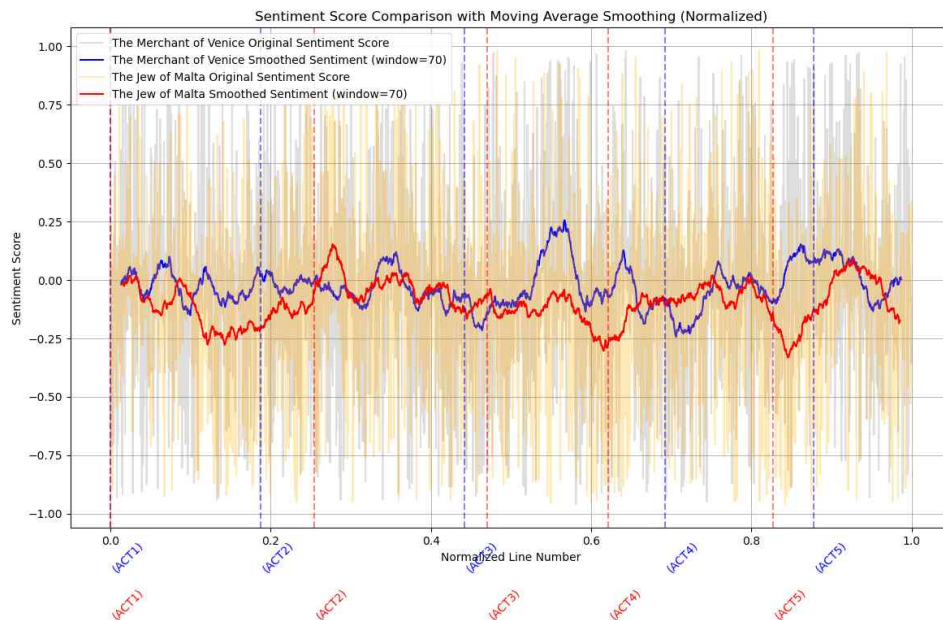
b. 맥베스 vs 템벌레인 대왕



c. 베니스의 상인 vs 유대인의 비극

위의 세 작품의 감정 변화 양상을 비교 결과, 두 작가 간의 서사 구조적 유사성이 뚜렷하게 나타나지는 않는다. 감정 변화의 폭이 나타나는 구간이 몇 군데 일치하는 곳이 있지만, 해당 결과만으로 셰익스피어가 크리스토퍼 말로우의 영향을 받았는지는 단언하기 어렵다. 이

는 작품 선정의 문제일 수 있으며 더 많은 작품들 간의 비교 분석을 통해 더 확실해질 수 있을 것이다.



4) 셰익스피어와 크리스토퍼 말로우의 문장 유사성 비교

각 작가별 작품을 취합해 문장 유사성을 검증해 임계값이 0.7 이상인 데이터를 추출하였다. 셰익스피어의 작품들을 기준으로 크리스토퍼 말로우의 작품 대사 중 유사성 점수가 0.7 이상일 경우인 값을 확인하였다. 결과 1) 유대인의 비극, 2364개의 대사 중 135개, 2) 템벌레인 대왕, 4891개 대사 중 244개, 3) 파우스트 박사의 비극, 1424개 대사 중 63개의 데이터가 집계되었다. 세 작품 모두 4~5% 비율로 낮은 유사도 비율을 보여주었으며, 유사성이 높게 나타난 데이터 중 중복된 대사, 'My Lord', 'Your Majesty' 등의 왕실 예의 표현 문장을 제외할 경우 그 비율은 더 낮아지게 된다. 즉, 기준으로 선정된 작품을 기준에서 텍스트의 일치성만으로는 문장 유사도만으로는 크리스토퍼 말로우와 셰익스피어의 유사성을 확인하기 힘들다.

다만, 그 외 문장들을 확인 시 일부 문장은 표현법 등이 유사한 문장도 존재했기에, 불필요한 표현 등을 제거하거나, 텍스트 자체가 아닌 주위 단어들과의 관계를 고려해 유사성을 확인한다면 두 작가의 문체에서의 유사성을 확인할 수 있을 가능성이 있다.

| | |
|--|--|
| text(christopher) | match sentence |
| Brother, I see your meaning well enough, | Come, come, we know your meaning, brother |
| But I refer me to my noblemen, | But you, my noble lords, may name the time; |
| Meander, thou, my faithful counsellor, | What, dost thou scorn me for my gentle counsel? |
| Oft have I heard your majesty complain | The news I have to tell your majesty |
| Full true thou speak'st, and like thyself, my lord, | Thou speak'st it well. Go, father, with thy son. |
| Full true thou speak'st, and like thyself, my lord, | What think'st thou? is it not an easy matter |
| Full true thou speak'st, and like thyself, my lord, | As thou and I; who, as thou know'st, are dear |
| Full true thou speak'st, and like thyself, my lord, | Thou speak'st with all thy wit: and yet, I' faith, |
| Full true thou speak'st, and like thyself, my lord, | These evils thou repeat'st upon thyself |
| Full true thou speak'st, and like thyself, my lord, | If thou speak'st false, |
| Doubt not, my lord and gracious sovereign, | My gracious sovereign? |
| Then now, my lord, I humbly take my leave. | So humbly take my leave. |
| To hear the king thus threaten like himself! | What! threat you me with telling of the king? |
| Since Fortune gives you opportunity | Since this fortune falls to you, |
| And vow to wear it for my country's good, | Or wear it on my sword, yet my poor country |
| I know it well, my lord, and thank you all. | I thank you, good my lord; and thank you all. |
| Sound up the trumpets, then. | Make all our trumpets speak; give them all breath, |
| God save the king! | God save the king! |
| God save the king! | God save the king! |
| Or look you I should play the orator? | Fear not, my lord, I'll play the orator |
| Come, let us march. | Well, march we on, |
| I hear them come: shall we encounter them? | Shall we well meet them; that way are they coming. |
| That I shall be the monarch of the East, | You shall be king. |
| Where kings shall crouch unto our conquering swords, | The sword of our slain kings: yet do not fear; |
| And now, fair madam, and my noble lords, | And for your grace; and you, my noble lords. |
| And now, fair madam, and my noble lords, | But you, my noble lords, may name the time; |
| Shall be my regent, and remain as king. | You shall be king. |

▲빨간색 표시: 문체, 표현 기법이 유사한 대사 / 파란색 표시: 유사성이 무의미한 대사

```
import pandas as pd
from sentence_transformers import SentenceTransformer, util

# 모델 불러오기
model = SentenceTransformer('sentence-transformers/all-MiniLM-L6-v2')

# CSV 파일 불러오기
marlowe_df = pd.read_csv("C:\\Users\\kang8\\OneDrive\\바탕 화면\\빅데이터융합개론\\크리스토퍼 말로우\\파우스트 박사\\RoBERTA_output_with_sentiment_scores.csv", encoding='cp949')
shakespeare_df = pd.read_csv("C:\\Users\\kang8\\OneDrive\\바탕 화면\\빅데이터융합개론\\셰익스피어\\archive\\shakespeare_plays_part.csv", encoding='utf-8')

# 결과 저장용 리스트 준비
results = []

# 셰익스피어 텍스트 임베딩 (전체 문장을 한 번에 임베딩하여 효율성 향상)
shakespeare_sentences = shakespeare_df['text'].tolist()
shakespeare_embeddings = model.encode(shakespeare_sentences, convert_to_tensor=True)

# 마로우의 각 문장을 셰익스피어 문장들과 비교
for marlowe_sentence in marlowe_df['text']:
    # 마로우 문장 인코딩
    marlowe_embedding = model.encode(marlowe_sentence, convert_to_tensor=True)

    # 셰익스피어 문장들과 유사도 계산
    similarities = util.cos_sim(marlowe_embedding, shakespeare_embeddings)[0]

    # 유사도가 임계값(0.7) 이상인 경우 결과 저장
    for idx, similarity in enumerate(similarities):
        if similarity > 0.7:
            results.append({'text(christopher)': marlowe_sentence, 'match sentence': shakespeare_sentences[idx]})

# 결과를 DataFrame으로 변환
results_df = pd.DataFrame(results)

# CSV 파일로 저장
results_df.to_csv("C:\\Users\\kang8\\OneDrive\\바탕 화면\\빅데이터융합개론\\크리스토퍼 말로우\\marlowe_shakespeare_similarity2.csv", index=False, encoding='utf-8-sig')
print("유사도 분석 완료. 결과가 'marlowe_shakespeare_similarity2.csv'에 저장되었습니다.")
```

▲문장 유사성 확인에 사용한 코드

4. 머신러닝 모델 적용

가. 사용된 모델

1) RoBERTa:

RoBERTa는 머신러닝 모델 중 하나로, 자연어 처리(NLP) 작업을 위해 설계된 사전 훈련된 언어 모델이다. RoBERTa는 BERT(Bidirectional Encoder Representations from Transformers)의 변형 모델로, 더 많은 데이터와 더 긴 훈련 시간, 다양한 훈련 기법을 사용하여 성능을 향상시킨 모델. 이 모델은 문맥을 이해하고 텍스트의 의미를 파악하는 데 강력한 능력을 가지고 있어 감정 분석 작업에 적합하다. (처음엔 감정 분석을 위해 VADER를 사용했지만, RoBERTa가 더 세부적인 감정 수치를 계산해 줄 수 있기에 모델을 변경함.)

2) LDA(Latent Dirichlet Allocation):

LDA는 주제 모델링 기법으로, 문서 집합에서 숨겨진 주제를 추출하는 데 사용된다. LDA에서 각 문서의 주제는 단어의 확률 분포로 표현된다. LDA는 반복적인 과정을 통해 각 단어와 주제 간의 관계를 학습하며, 이 결과로 주제에 대한 단어들을 추출하여 주제를 해석할 수 있다.

3) sentence-transformers/all-MiniLM-L6-v2

자연어 처리를 위한 Transformer 기반 모델로, 특히 문장 간 유사도 분석을 위한 임베딩을 생성하는 데 사용된다. 각 문장을 고정된 길이의 벡터로 변환하는 방식으로 작동하며, 이를 통해 유사도 분석, 검색, 군집화 등의 작업에 유용한 문장 임베딩을 생성 가능하다.

a. 결과 지표

LDA 모델의 성능 지표는 Perplexity와 Coherence로 구분된다.

I) **Perplexity**: 모델이 특정 단어가 주제 내에서 나올 확률을 얼마나 잘 예측하는지

II) **Coherence**: 각 주제에 포함된 단어들이 서로 연관성이 높고 논리적으로 연결되어 있는지 표현

- 희극: Perplexity: -9.899027737875793, Coherence Score: 0.3093564756400327

- 역사극: Perplexity: -9.777251869470444, Coherence Score: 0.407917055049783

- 비극: Perplexity: -9.803674087546066, Coherence Score: 0.43781200893380107

5. 결론 및 제안

가. 결론

본 연구는 셰익스피어의 극 작품들을 장르별로 나누어 감정분석을 통해 이야기의 서사 구조를 분석하였다. 또한 작가와 비교되는 크리스토퍼 말로우의 작품을 선정해 셰익스피어의 이야기와 감정분석을 통해 두 작품의 유사성에 대해 알아보았다. 셰익스피어 자신의 작품들 간에서 서사에 따른 감정 변화에는 유의미한 유사성과 구별점이 확인되었고 주제어 분

석을 통해 장르별로 구분되는 주제어가 존재하는지 확인했다.

마지막으로 크리스토퍼 말로우와의 비교 분석을 진행했지만, 그 유사성이 희미했다. 이는 작품 쌍 선정 등의 문제가 있을 수 있으며, 더 많은 양의 데이터와 비교분석을 위한 세부적인 고려요소가 필요하다.

나. 제안

본 연구는 문학 작품을 시각적으로 그 서사 구조를 확인함에 의의가 있으며, 문학 작품 해석에 대해 데이터 분석을 통해 접근한 방법이다. 텍스트 분석을 통해 작가 미상의 옛 고전 문학 작품들에 대한 구조를 분석하고 유사한 구조를 가진 작품의 작가를 확인하는 등의 방식으로 발전될 수 있다.