aiffel-mainquest, 2024, 1–3
doi: 000-00000
Fake Article

Assessing Bias in AI-Generated Word Embeddings: A WEAT Analysis of Korean Film Genres

HyoungChul Park[1,*]

[1]Aiffel Research Online-9th, Park, Yeoksam, 05514, Seoul, Korea

*Corresponding author. email-id.com

FOR PUBLISHER ONLY Received on Date Month Year; revised on Date Month Year; accepted on Date Month Year

Abstract

As AI systems are increasingly used in decision-making, concerns about biases in training data have grown. Instances of AI algorithms favoring certain demographic groups highlight the potential for biased outcomes. The Word Embedding Association Test (WEAT) offers a way to quantitatively assess biases in word embeddings. In this study, we applied WEAT to measure biases in word embeddings from Korean film synopses, classified into commercial or artistic genres. We compared WEAT scores across different preprocessing methods applied to representative word datasets to assess bias. Our findings contribute to understanding biases in genre-based content and offer insights for ethical AI development in content analysis.

Key words: Word2Vec, Embedding, WEAK, Bias

Introduction

In recent years, cases have emerged where AI algorithms in hiring processes have favored male candidates over female candidates, or in judicial sentencing, assigned harsher penalties to Black individuals than to white individuals. These examples highlight how bias embedded within training datasets can significantly impact the performance and fairness of AI models. As a result, growing concerns around AI accountability have spurred efforts to mitigate bias present in datasets.

In this study, we employed the Word Embedding Association Test (WEAT) to quantitatively measure biases embedded in word embedding vectors trained on various datasets. Specifically, we investigated whether biases exist in Korean film synopses data, where films are categorized as either commercial or artistic based on genre. The experiment involved a qualitative comparison of WEAT scores across different preprocessing strategies applied to representative word datasets, to evaluate bias presence.

Method

To analyze bias within the data, we first selected representative words from the dataset and converted them into vectorized forms suitable for model training. We then conducted a quantitative assessment of the biases embedded within vector similarities. This analysis serves as a foundational step toward identifying and mitigating biases in model training.

Morphological Analysis - konlpy

Morphological analysis is a linguistic process that breaks down words into their basic components, such as roots, prefixes, and suffixes, to understand their meanings and roles within sentences. In this study, we used konlpy for morphological analysis to extract nouns from our dataset, allowing us to focus on key terms for a more efficient and targeted analysis.

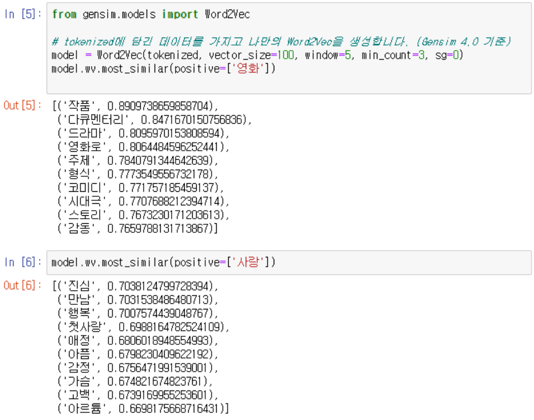
Word Embedding - Word2Vec

For word embedding, we employed the Word2Vec model to represent words in the dataset as continuous vector representations, capturing semantic relationships between them.

Word Embedding Association Test (WEAT)

The Word Embedding Association Test (WEAT) was developed as a tool to analyze and measure biases embedded within word embeddings. Originating from concerns over how machine learning models capture and potentially reinforce societal biases, WEAT provides a quantitative approach to evaluating these biases in word embeddings like Word2Vec. This test is essential because, as machine learning systems increasingly influence various aspects of daily life, understanding and addressing embedded biases in language representations has become a critical aspect of ethical AI development.

WEAT operates by measuring the association strength between sets of target words (e.g., “career” vs. “family”) and attribute words (e.g., “male” vs. “female”). Using a similarity metric (typically cosine similarity) within the embedding space, WEAT



```
In [5]: from gensim.models import Word2Vec

        # tokenized에 담긴 데이터를 가지고 나만의 Word2Vec을 생성합니다. (Gensim 4.0 기준)
        model = Word2Vec(tokenized, vector_size=100, window=5, min_count=3, sg=0)
        model.wv.most_similar(positive=['영화'])

Out[5]: [('작품', 0.8909738659858704),
         ('다큐멘터리', 0.8471670150756836),
         ('드라마', 0.8095970153808594),
         ('영화로', 0.8064484596252441),
         ('주제', 0.7840791344642639),
         ('형식', 0.7773549556732178),
         ('코미디', 0.7717571854591 37),
         ('시대극', 0.7707688212394714),
         ('스토리', 0.7673230171203613),
         ('감동', 0.7659788131713867)]

In [6]: model.wv.most_similar(positive=['사랑'])

Out[6]: [('진심', 0.7038124799728394),
         ('만남', 0.7031538486480713),
         ('행복', 0.7007574439048767),
         ('첫사랑', 0.6988164782524109),
         ('애정', 0.6806018948554993),
         ('아픔', 0.6798230409622192),
         ('감정', 0.6756471991539001),
         ('가슴', 0.674821674823761),
         ('고백', 0.6739169955253601),
         ('아르툼', 0.6698175668716431)]
```

Fig. 1: The figure below illustrates the use of Word2Vec to represent words as embedding vectors, followed by similarity measurements to identify the most similar words for a given word. The upper part shows similarity analysis for the word ”movie,” while the lower part focuses on similarity analysis for the word ”love.”

evaluates whether certain target words align more closely with one set of attribute words than the other. The test generates an effect size that quantifies the bias level. Higher effect sizes indicate stronger associations, revealing potential biases embedded within the word embeddings.

$$\text{WEAT Score} = \frac{\text{mean}_{x \in X}\, s(x, A, B) - \text{mean}_{y \in Y}\, s(y, A, B)}{\text{std}_{w \in X \cup Y}\, s(w, A, B)} \quad (1)$$

where,

$$s(w, A, B) = \text{mean}_{a \in A} \cos(\vec{w}, \vec{a}) - \text{mean}_{b \in B} \cos(\vec{w}, \vec{b}) \quad (2)$$

A, B are sets of attributes, X,Y are sets of targets

Remove duplicate word for attribute, target set

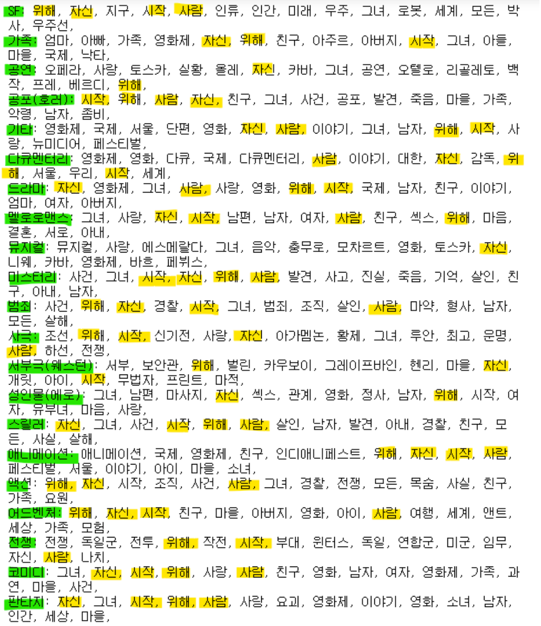
To construct the attribute and target sets for WEAT, we first extracted representative words by applying the Term Frequency-Inverse Document Frequency (TF-IDF) method. TF-IDF is a technique commonly used in natural language processing to evaluate the importance of a word within a specific document relative to a larger set of documents. It combines two metrics: term frequency (TF), which measures the frequency of a word within a document, and inverse document frequency (IDF), which discounts words commonly appearing across all documents. By prioritizing words with high TF-IDF scores, we identified the 15 most relevant terms, which served as the foundation for our WEAT sets. Duplicate words were then manually removed and replaced with other suitable terms to refine the final sets.

Result

Dataset Description:

The dataset used in this experiment contains synopsis information for films produced from 2001 to August 2019.

Film Categories



SF: 위해, 자신, 지구, 시작, 사람, 인류, 인간, 미래, 우주, 그녀, 로봇, 세계, 모든, 박사, 우주선,
가족: 엄마, 아빠, 가족, 영화제, 자신, 위해, 친구, 아주르, 아버지, 시작, 그녀, 아들, 마을, 국제, 낙타,
공연: 오페라, 사랑, 토스카, 실황, 올레, 자신, 카바, 그녀, 공연, 오텔로, 리골레토, 백작, 프레, 베르디, 위해,
공포(호러): 시작, 위해, 사람, 자신, 친구, 그녀, 사건, 공포, 발견, 죽음, 마을, 가족, 악령, 남자, 좀비,
기타: 영화제, 국제, 서울, 단편, 영화, 자신, 사람, 이야기, 그녀, 남자, 위해, 시작, 사랑, 뉴미디어, 페스티벌,
다큐멘터리: 영화제, 영화, 다큐, 국제, 다큐멘터리, 사람, 이야기, 대한, 자신, 감독, 위해, 서울, 우리, 시작, 세계,
드라마: 자신, 영화제, 그녀, 사람, 사랑, 영화, 위해, 시작, 국제, 남자, 친구, 이야기, 엄마, 여자, 아버지,
멜로로맨스: 그녀, 사랑, 자신, 시작, 남편, 남자, 여자, 사람, 친구, 섹스, 위해, 마음, 결혼, 서로, 아내,
뮤지컬: 뮤지컬, 사랑, 에스메랄다, 그녀, 음악, 충무로, 모차르트, 영화, 토스카, 자신, 니웨, 카바, 영화제, 바흐, 페뷔스,
미스터리: 사건, 그녀, 시작, 자신, 위해, 사람, 발견, 사고, 진실, 죽음, 기억, 살인, 친구, 아내, 남자,
범죄: 사건, 위해, 자신, 경찰, 시작, 그녀, 범죄, 조직, 살인, 사람, 마약, 형사, 남자, 모든, 살해,
사극: 조선, 위해, 시작, 신기전, 사랑, 자신, 아가멤논, 황제, 그녀, 루안, 최고, 운명, 사람, 하선, 전쟁,
서부극(웨스턴): 서부, 보안관, 위해, 벌린, 카우보이, 그레이프바인, 헨리, 마을, 자신, 개릿, 아이, 시작, 무법자, 프린트, 마적,
성인물(에로): 그녀, 남편, 마사지, 자신, 섹스, 관계, 영화, 정사, 남자, 위해, 시작, 여자, 유부녀, 마음, 사랑,
스릴러: 자신, 그녀, 사건, 시작, 위해, 사람, 살인, 남자, 발견, 아내, 경찰, 친구, 모든, 사실, 살해,
애니메이션: 애니메이션, 국제, 영화제, 친구, 인디애니페스트, 위해, 자신, 시작, 사람, 페스티벌, 서울, 이야기, 아이, 마을, 소녀,
액션: 위해, 자신, 시작, 조직, 사건, 사람, 그녀, 경찰, 전쟁, 모든, 목숨, 사실, 친구, 가족, 요원,
어드벤처: 위해, 자신, 시작, 친구, 마을, 아버지, 영화, 아이, 사람, 여행, 세계, 앤트, 세상, 가족, 모험,
전쟁: 전쟁, 독일군, 전투, 위해, 작전, 시작, 부대, 윈터스, 독일, 연합군, 미군, 임무, 자신, 사람, 나치,
코미디: 그녀, 자신, 시작, 위해, 사랑, 사람, 친구, 영화, 남자, 여자, 영화제, 가족, 과연, 마을, 사건,
판타지: 자신, 그녀, 시작, 위해, 사람, 사랑, 요괴, 영화제, 이야기, 영화, 소녀, 남자, 인간, 세상, 마을,

Fig. 2: The following figure displays the extracted words for the attribute and target sets in WEAT. Words highlighted in yellow represent terms that appeared redundantly across multiple categories and were subsequently removed.

Genre			
Science Fiction	Family	Performing Arts	Mystery
Horror	Documentary	Drama	Crime
Romance	Musical	Fantasy	Historical
Western	Adult (Erotic)	Thriller	Animation
Action	Adventure	War	Comedy
Others			

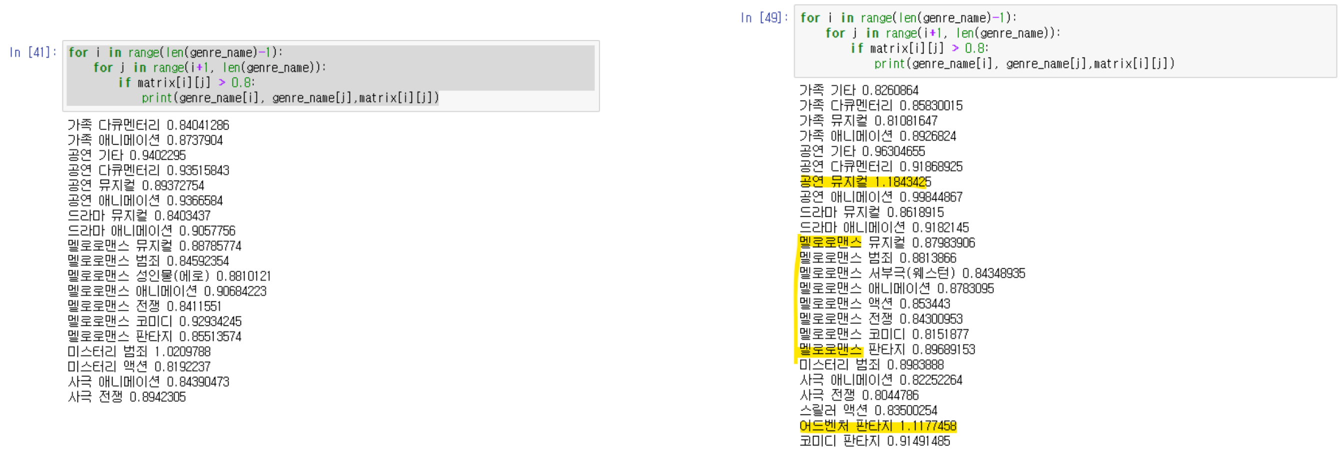
synopsis_art.txt: Art Films synopsis_gen.txt: General Films (Commercial Films) Other films are categorized as Independent Films, etc. Genre Categories

After refining the dataset, we analyzed whether commercial and art films exhibit genre-based biases. In Figure 3, pairs of genres with WEAT scores exceeding 0.8 are displayed, indicating a strong bias, with the first genre perceived as more ”artistic” and the second as more ”commercial.” From a qualitative analysis of the results, melodrama/romance appeared to be perceived as more aligned with art films compared to other genres.

Even after word processing, this tendency remained consistent, suggesting that outlier words in the word set used for calculating WEAT scores did not significantly impact the final results. Since the WEAT score is an average-based value, it seems to normalize the effects of these outliers.

Conclusion

The existence of methodologies for detecting bias within datasets is promising. However, the results of bias analysis can vary significantly depending on which words are included in the model. Additionally, distinguishing between biases that are socially



```
In [41]: for i in range(len(genre_name)-1):
             for j in range(i+1, len(genre_name)):
                 if matrix[i][j] > 0.8:
                     print(genre_name[i], genre_name[j],matrix[i][j])
```

```
가족 다큐멘터리 0.84041286
가족 애니메이션 0.8737904
공연 기타 0.9402295
공연 다큐멘터리 0.93515843
공연 뮤지컬 0.89372754
공연 애니메이션 0.9366584
드라마 뮤지컬 0.8403437
드라마 애니메이션 0.9057756
멜로로맨스 뮤지컬 0.88785774
멜로로맨스 범죄 0.84592354
멜로로맨스 성인물(에로) 0.8810121
멜로로맨스 애니메이션 0.90684223
멜로로맨스 전쟁 0.8411551
멜로로맨스 코미디 0.92934245
멜로로맨스 판타지 0.85513574
미스터리 범죄 1.0209788
미스터리 액션 0.8192237
사극 애니메이션 0.84390473
사극 전쟁 0.8942305
```

```
In [49]: for i in range(len(genre_name)-1):
             for j in range(i+1, len(genre_name)):
                 if matrix[i][j] > 0.8:
                     print(genre_name[i], genre_name[j],matrix[i][j])
```

```
가족 기타 0.8260864
가족 다큐멘터리 0.85830015
가족 뮤지컬 0.81081647
가족 애니메이션 0.8926824
공연 기타 0.96304655
공연 다큐멘터리 0.91868925
공연 뮤지컬 1.1843425
공연 애니메이션 0.99844867
드라마 뮤지컬 0.8618915
드라마 애니메이션 0.9182145
멜로로맨스 뮤지컬 0.87983906
멜로로맨스 범죄 0.8813866
멜로로맨스 서부극(웨스턴) 0.84348935
멜로로맨스 애니메이션 0.8783095
멜로로맨스 액션 0.853443
멜로로맨스 전쟁 0.84300953
멜로로맨스 코미디 0.8151877
멜로로맨스 판타지 0.89689153
미스터리 범죄 0.8983888
사극 애니메이션 0.82252264
사극 전쟁 0.8044786
스릴러 액션 0.83500254
어드벤처 판타지 1.1177458
코미디 판타지 0.91491485
```

Fig. 3: The figure above shows the results of the WEAT word set before and after preprocessing, displaying only target pairs with scores above 0.8.

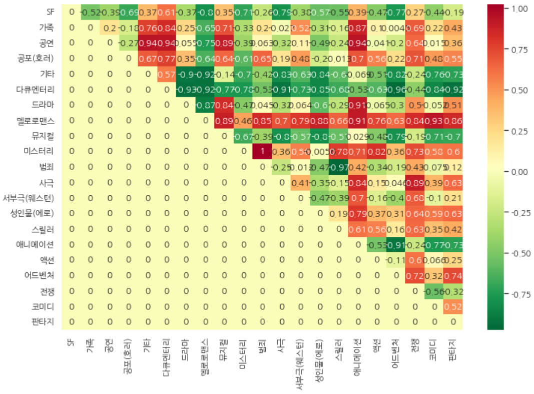


Fig. 4: The figure above displays a heatmap of all genre pairs. Higher scores indicate a stronger bias toward "art" for the row genre and "commercial" for the column genre, while lower scores suggest the opposite.

acceptable and those that are not is a matter that extends beyond the realm of engineering. For example, biases associated with pleasant or unpleasant words may be deemed acceptable, while biases linked to race, gender, or similar attributes are generally not tolerated in society.

Bias is not a static concept; it shifts with societal norms over time. Thus, continually updating and applying standards to address acceptable and unacceptable biases remains a critical aspect of this task. Furthermore, the challenge does not end with identifying biases, as the process of mitigating or removing them presents an additional, complex challenge.

Finally, limitations also arise from the use of Word2Vec for word embeddings, as homonyms are not distinguished within this framework, potentially introducing additional ambiguity into the analysis.

References