김종윤
2023-28318
협동과정 인공지능 전공

소셜컴퓨팅

과제 1 - 2주차

1. 키워드 선택 이유

이번주 과제로 선택한 키워드는 “애플페이”, “OpenAI”이다. “애플페이”가 3월 21일에 서비스를 시작한다는 소문으로 트위터내에 다양한 트윗이 게시되었을 것으로 생각되어 키워드로 선택했다. “OpenAI”는 ChatGPT를 만들어 서비스하는 기관의 이름이다. 키워드를 구글 시트에 작성하는 날, GPT-4가 발표되어 많은 사람들의 관심이 트윗으로 전해졌을 것으로 생각되어 키워드로 선택했다.

2. 트윗 데이터의 구성

트윗 데이터를 분석하기에 앞서 데이터 (.json file) 의 structure를 알고 있어야한다. 트위터 개발자계정이 제대로 생성되어서 API 문서를 직접 조회할 수 있었다면 API의 response 구조와 데이터 타입, Optional 여부 등을 알고 python dataclass로 만들어 빠르게 파싱할 수 있었겠지만 아직 개발자계정 생성이 완료되지 않은 점이 조금 아쉽다.

제출한 코드 directory/week2에 보면 데이터 구조를 이해하기 위해 dataclass를 직접 정리해둔 파이썬 파일이 있다. 이 파일을 기반으로 데이터 분석을 시도했다.

3. 트윗 기초 통계 분석

3.1.언제 트윗되었는 가?

Figure 1을 보면 트윗 생성된 시간을 그래프로 볼 수 있다. 애플페이의 경우 3/13일 오전 7시 부터 3/15일 새벽 5시까지 트윗이 생성되었고, OpenAI의 경우, 3/15일 오후 2시부터 같은 날 오후 3시까지 트윗이 생성되었다.

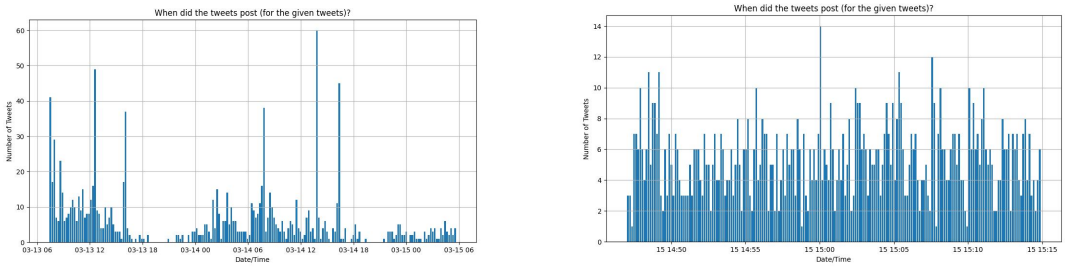

Figure 1: 트윗 생성시간 (좌: 애플페이, 우: OpenAI, X축: 시간, Y축: 트윗 개수)

트윗 개수를 1100개로 제한한 이유로 해당 트윗이 얼마나 더 화제가 되었는 지도 알 수 있다. OpenAI의 경우 약 13시간 만에 1100개 가량의 트윗이 생성되었지만 애플페이는 약 46시간 정도의 시간이 필요했다.

즉, OpenAI가 더 화제성이 좋은 키워드이다. 애플페이의 경우 한국시간 기준 유저 활동량이 많은 시간대에 (오후 3시 ~ 9시, UTC기준 6시~12시) 트윗이 많이 되었다.

3.2.언제 계정이 생성되었는 가?

애플페이 키워드를 트윗한 많은 계정들이 2022년 이후에 생성된 반면, OpenAI를 트윗한 계정들의 경우 2010년 전후에 생성, 2022년 이후에 많이 생성되었다. 또한 OpenAI 키워드를 트윗한 계정들의 분포가 2010년부터 꾸준하게 나타나는데 이는 단순히 해당 화제성 키워드를 이용해 광고나 어뷰징을 목적으로 한 계정이 적다고 생각해볼 수도 있다.

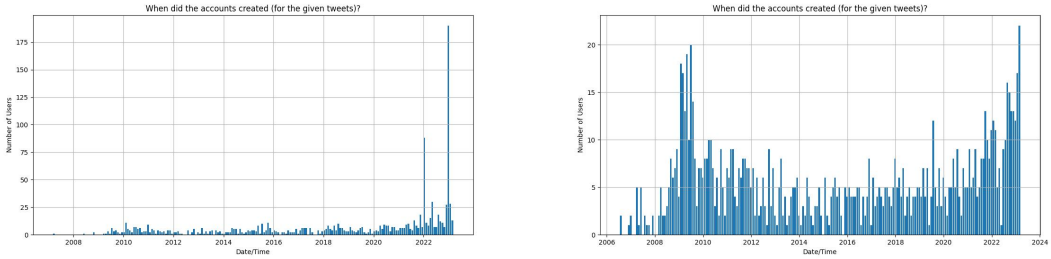

Figure 2: 키워드를 트윗한 계정 생성 날짜
(좌: 애플페이, 우: OpenAI, X축: 시간, Y축: 계정 개수)

3.3.몇 개나 트윗했는 가?

두개 키워드 모두 많은 유저들이 트윗을 많이 하지는 않았다. OpenAI 키워드를 트윗한 경우는 비교적 덜 가파른 power-law graph를 보인다. Power-law를 따른다고 생각하고 log-log graph도 그려보았다 (figure4). log-log 그래프의 경우, 어느 수준 (트윗 개수 약 10^4) 까지는 증가하다 감소한다. (이때, 트윗개수가 0개로 수집된 데이터의 경우, log-scale에 그릴수 없기 때문에 제거되었다.) log-log scale 그래프가 강의시간에 본 그래프의 형태를 띈다는 점에서 놀라웠다.

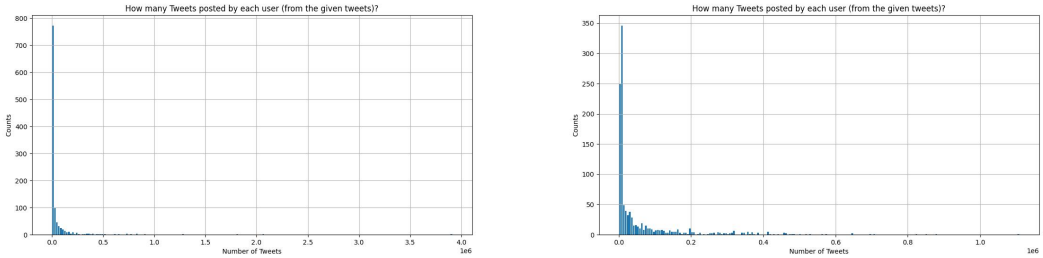

Figure 3: 키워드를 트윗한 계정의 트윗 개수
(좌: 애플페이, 우: OpenAI, X축: 트윗 개수, Y축: 유저 수)

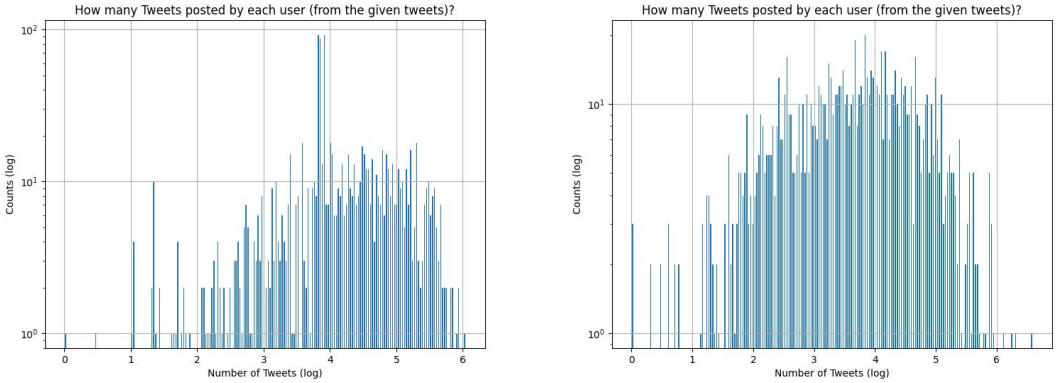

Figure 4: 키워드를 트윗한 계정의 트윗 개수
(좌: 애플페이, 우: OpenAI, X축: 트윗 개수(log), Y축: 유저 수(log))

3.4.수집된 tweet 중 가장 많은 팔로워를 가진 사용자는?

우선, 팔로워 수 - 유저수 그래프를 그려보았다. 예상대로 많은 유저들이 팔로워수가 많지 않았다. 마찬가지로 power-law를 따른다고 생각해 log-log scale 그래프 (Figure 6)를 그려보았다. 3.3에서의 분석과 비슷하게 100~1000명에서 최대, 그 이후 감소하는 모습을 보였다.

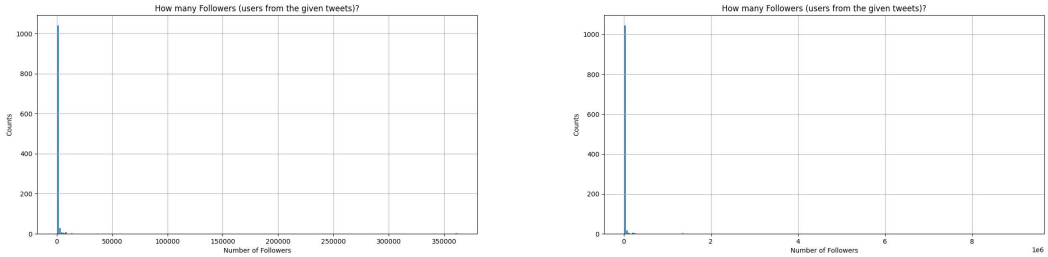

Figure 5: 키워드를 트윗한 계정의 트윗 개수
(좌: 애플페이, 우: OpenAI, X축: 팔로워 수, Y축: 유저 수)

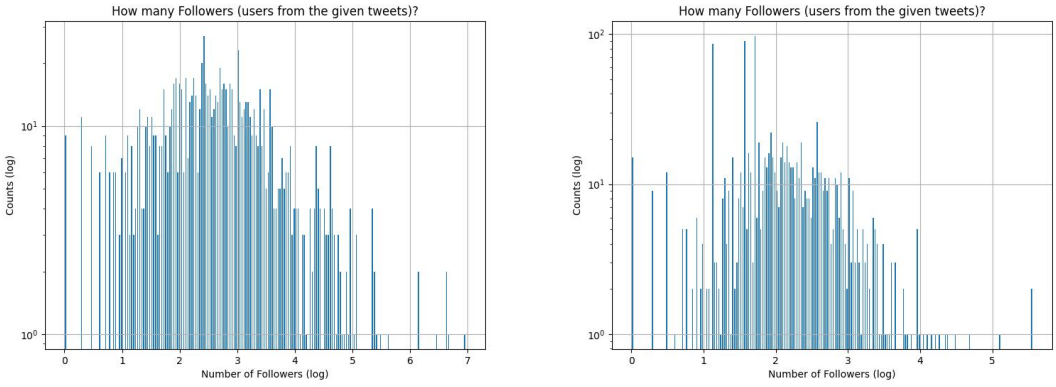

Figure 6: 키워드를 트윗한 계정의 트윗 개수
(좌: 애플페이, 우: OpenAI, X축: 팔로워 수, Y축: 유저 수) (log-log scale)

가장 많은 팔로워를 가진 사용자는 애플페이 키워드에서는 약 36만 팔로워를 가진 "YTN" (공식계정) 그리고 그 뒤로는 약 13만명 팔로워를 가진 "PGM"이라는 계정과 약 4.7만 팔로워의 "머니투데이" 계정이 있었다. OpenAI 키워드의 경우, 920만 팔로워의 "Bloomberg" (공

식계정), 477만 팔로워의 “NTN24” (공식계정, 콜롬비아 방송사) 그리고 432만 팔로워의 “Economics Times” (공식계정) 순이었다.

3.5.가장 많은 hashtag는

애플페이 키워드에서 가장 많은 해쉬태그는 14번 언급된 “캐럿들은_언제나_세븐틴 편”, 9번 언급된 “애플페이”, 8번 언급된 “애플” 순 이었다. “캐럿들은_언제나_세븐틴편” 라는 키워드가 14번 언급된 것은 앞서 3.2에서 분석한대로 2022년 직후에 최근 생성 계정이 많아 어뷰징이 많을 것 같다는 생각을 지지한다. 애플페이라는 화제성 키워드에 다른 해쉬태그를 넣어 다른 목적으로 트윗이 많이 노출되게끔했다라는 결론을 낼 수 있다.

OpenAI 키워드에서 가장 많은 해쉬태그는 “OpenAI” 35회, “GPT4” 34회, “ChatGPT” 33회로 예상대로 기관의 이름, 최근 발표된 모델인 GPT4, 그리고 화제의 GPT 서비스인 ChatGPT 순으로 해쉬태그가 많이 사용되었다.

3.6.트윗 본문(text) 빈출 키워드(명사)는?

애플페이 키워드에서는 “애플” 1202회, “페이” 1142회 순으로 예상대로 애플페이를 서비스하는 애플과 애플페이를 띄어쓴 애플과 페이가 가장 많이 사용되었다. 눈여겨 볼만한 다른 명사로는 “쿠팡” 272회, “파트너” 272회, “활동” 271회, “일환” 271회, “정액” 268회, “수수료” 261회가 있다. 앞서 언급한 단어들은 광고의 목적으로 사용될 수 밖에 없는데 이는 3.2에서 분석한 “어뷰징과 광고 목적의 트윗이 많다”라는 주장을 뒷받침한다.

OpenAI 키워드에서는 “@“ 988회, “https” 704회, “OpenAI” 662회, “RT” 614회, “GPT-4” 416회, “ChatGPT” 167회, “AI” 152회, “model” 133회 순이다. Stopwords 에 @, https, rt 등 트위터 특유의 symbol과 단어를 넣지 못했고 nltk 분석기가 위 단어들을 명사라 분석해 통계에 함께 나타났다. 하지만 이는 누군가를 mention하면서 링크를 경우가 많다는 것을 그리고 RT도 많이 되었음도 볼 수 있는 좋은 통계이다. 나아가 예상한 대로 OpenAI, GPT-4, ChatGPT가 가장 많이 본문에서 사용된 명사였다. 트윗의 주제가 인공지능인 만큼 그 이후로는 AI, model이 이어졌다.

3.7.(+α) 트윗, 유저 언어 통계

추가로 트윗의 화제가 된 지역을 알아보기 위해 트윗과 트윗을 게시한 유저의 언어 통계를 만들어보았다.

애플페이 키워드는 모든 트윗이 한국어로 작성되었고, 유저의 언어 설정은 모두 None 이었다. “애플페이”라는 단어 자체가 한글이기 때문에 당연한 결과였다. 유저의 언어설정이 없는 이유는 계정 생성시 입력하지 않은 것으로 생각된다.

OpenAI 키워드는 734개의 트윗이 영어, 128개의 트윗이 일본어로 적혀있었다. OpenAI가 미국에 있는 점 그리고 많은 인공지능 관련 내용들이 영어로 되어 있다는 점이 영어 트윗이 많음을 추측할 수 있었다. 다만 일본어가 그 다음을 이은 것이 의아했다. 유저 언어설저은 마찬가지로 모두 None이다.

3.8.리트윗(RT) 통계

애플페이 키워드 트윗 중 top-10 RT 트윗은 다음과 같다. (트윗 내용은 길이를 줄여 출력되었다.) 1541번 RT된 “zzenmekeneng” 유저의 트윗이 가장 많이 RT 되었다. 놀라운 점은 가장 팔로워가 많았던 YTN 트윗과 머니투데이의 트윗은 많이 RT되지 않았다는 것이다.

[{'RT count': 1541,
 'Tweet': 'RT @zzenmekeneng: 근데 현대에서 애플페이 1년 독점한다고 신규고객이 많이 늘을까?? ...'
 'User': 'zzenmekeneng'},
{'RT count': 729,
 'Tweet': "RT @lucripeta: [단독] 'D-8일'…애플페이, 21일부터 한국 서비스 개시..."
 'User': 'lucripeta'},
{'RT count': 582,
 'Tweet': 'RT @sun1it: 애플페이 출시 직후는 현대카드만 되고 교통카드랑 타 카드사는 추후 가능해지나바...'
 'User': 'sun1it'},
{'RT count': 523,
 'Tweet': 'RT @YUNHO1812: 애플페이 가맹점수 일주일도 안되어서 확 늘었네요'
 'User': 'YUNHO1812'},
{'RT count': 431,
 'Tweet': 'RT @t_ransborder: 애플페이, 이달 하순부터 국내 사용 가능'
 'User': 't_ransborder'},
{'RT count': 416,
 'Tweet': 'RT @Upgrade_MyLife: 카카오페이에서 애플티비 플러스, 애플 아케이드, 애플뮤직, ...'
 'User': 'Upgrade_MyLife'},
{'RT count': 326,
 'Tweet': 'RT @warden_america: [단독] ˙D-8일 ˚ …애플페이, 21일부터 한국 서비스 개시 (출처 :한국경제 | ...'
 'User': 'warden_america'},
{'RT count': 325,
 'Tweet': 'RT @warden_america: 배민에 떠버린 애플페이 결제란 배민 빠르다...! ...'
 'User': 'warden_america'},
{'RT count': 261,
 'Tweet': 'RT @dino_1155: 충격 실화 곧 애플페이 됩니다 https://t.co/jTfVH395VH',
 'User': 'dino_1155'},
{'RT count': 255,
 'Tweet': 'RT @kuromisia: 애플페이 스티커 붙었네 https://t.co/Ylpl4lVyTU',
 'User': 'kuromisia'}]

OpenAI 키워드 트윗 중 top-10 RT 트윗은 다음과 같다. (트윗 내용은 대부분 후략되었다.) 애플페이와 다르게 “OpenAI”에서 트윗한 GPT4 관련 트윗이 1.6만회로 가장 많이 트윗되었다. 그 이후에는 일본어 트윗과 스페인 트윗이 많이 RT된 10개 트윗에 나왔다는 점이 놀라운 점이다.

[{'RT count': 16356,
 'Tweet': 'RT @OpenAI: Announcing GPT-4, a large multimodal model, with our ...'
 'User': 'OpenAI'},
{'RT count': 4335,
 'Tweet': 'RT @sama: here is GPT-4, our most capable and aligned model yet. ...'
 'User': 'sama'},
{'RT count': 3187,
 'Tweet': 'RT @gomezidao: Gomezi Network: - large open source, - built in the ...'
 'User': 'gomezidao'},
{'RT count': 2173,
 'Tweet': "RT @unusual_whales: OpenAI's ChatGPT has reportedly predicted that ..."
 'User': 'unusual_whales'},
{'RT count': 2116,
 'Tweet': "RT @The_Delysium: @OpenAI Speaking of breakthroughs - here's the ..."
 'User': 'The_Delysium'},
{'RT count': 1716,
 'Tweet': 'RT @thealexbanks: OpenAI just launched GPT-4. ...'
 'User': 'thealexbanks'},
{'RT count': 1617,
 'Tweet': 'RT @kana_Eng_coach: 世界一人気の語学アプリDuolingoが、 OpenAI社のGPT-4...'
 'User': 'kana_Eng_coach'},
{'RT count': 1343,
 'Tweet': 'RT @gdb: We’re releasing GPT-4 — a large multimodal model (image ...'
 'User': 'gdb'},
{'RT count': 1067,
 'Tweet': "RT @spectatorindex: BREAKING: OpenAI's ChatGPT-4 passed the Law ..."
 'User': 'spectatorindex'},
{'RT count': 1052,
 'Tweet': 'RT @DotCSV: 🔴 GPT-4 YA ESTÁ AQUÍ!!! ...'
 'User': 'DotCSV'}]

RT된 트윗의 중복을 제거하기 위해 아래와 같은 코드를 작성했다. 우선, 중복을 모두 포함한 RT된 트윗을 리스트로 모았다. 이후, python dictionary의 key는 중복을 허용하지 않는 점을 이용해 RT의 id를 dictionary의 key로, (RT된 횟수, RT된 트윗의 유저 이름, RT된 트윗의 내용)으로 구성된 튜플을 value로 갖게 하였다. 이후, 내용의 중복 (e.g. 동일 내용으로 작성된 광고)를 제거하기 위해 RT 트윗의 내용을 key로 갖는 dictionary를 한번더 구성해 내용 중복을 제거하였다.

RT 트윗의 중복을 id와 내용으로 제거후, RT된 횟수로 정렬해 출력했다.

```python
# ============ RTd Tweets ============ #
def stat_retweets(self, ):
    retweeted_tweets = [status for status in self.twitter_data if 'retweeted_status' in status]
    print(f"중복 포함한 전체 RT 개수: {len(retweeted_tweets)}")

    # remove duplicates with dictionary
    retweets = {
        status["retweeted_status"]["id"]: (
            status['retweet_count'], status['retweeted_status']['user']['screen_name'], status['text'])
        for status in self.twitter_data if 'retweeted_status' in status
    }
    print(f"Tweet ID로 중복 제거한 RT 개수: {len(retweets)}")

    duplicate_content_removed_retweets = {
        retweet[1][2]: (retweet[1][0], retweet[1][1])
        for retweet in retweets.items()
    }
    print(f"내용 중복 제거한 RT 개수: {len(duplicate_content_removed_retweets)}")

    top10_rts = list(map(
        lambda i: {
            "User": i[1][1],
            "Tweet": i[0],
            "RT count": i[1][0]
        },
        sorted(
            duplicate_content_removed_retweets.items(),
            key=lambda rt: rt[1][0],
            reverse=True
        )[:10]
    ))
    print("================================")
    print("top-k retweeted Tweets (k=10)")
    pprint(top10_rts)
    print("================================")
```

Figure 7: RT된 트윗 분석 코드

과제에 작성한 모든 코드는 https://github.com/ArtemisDicoTiar/snu-social-computing 에서 확인할 수 있습니다.