



메타버스 내 익명 화자 인식 시스템

동국대학교 컴퓨터공학과
2019112130 조양진

목차

- 연구 목적 & 배경
- Contribution
- 수행 계획 및 방법
- 기술 선택 이유
- 기대 Output

연구 목적 & 배경

- 메타버스는 현실세계를 온라인 공간에 구현하여 사용자들이 아바타를 통해 사회적 활동을 할 수 있는 플랫폼
- 메타버스 내 범죄가 사각지대에 놓여있음
- 주요 메타버스 플랫폼인 로블록스, 제페토, 마인크래프트의 이용자는 대부분 10대이며, 제페토의 경우 글로벌 누적 이용자 2억 명 중 국내에서는 7~12살이 50.4%, 13~18살이 20.6%를 차지
- 로블록스도 유사하게 7~18살 이용자가 과반수
- 송혜진, 남완우 (2022) 메타버스 내 범죄발생 유형과 확장성에 관한 연구, 한국재난정보학회논문집, 18:1, 218-227

연구 목적 & 배경

- 메타버스 플랫폼(VRChat, Roblox 등)에서 익명성을 악용한 모욕, 성범죄, 사기 사례 증가 현재는 문제 사용자가 계정을 변경하면 신원 추적이 불가능한 상황
- 대부분의 범죄가 1:1 환경에서 발생하여 증인이 없고 증거 수집이 어려움
- 익명 환경에서 문제 사용자의 음성 특징을 기반으로 동일인 여부를 식별하는 시스템 개발 & 사용자 평가 시스템과 화자 인식 기술을 결합한 실시간 위험 사용자 경고 시스템 PoC 구현

Contribution

- 메타버스 환경에 특화된 화자 인식 응용 방안 제시
- 익명 평가와 화자 인식을 결합한 새로운 사용자 보호 메커니즘 제안
- 오픈소스 화자 인식 툴(ESPnet)의 메타버스 적용 사례 연구


수행 계획 및 방법

- 1단계: 음성 수집 및 특징 추출
 - ESPnet을 활용한 화자 임베딩 (speaker embedding) PoC 진행

```
# ESPnet을 활용한 화자 임베딩 추출 간략 코드
import torch
from espnet2.bin.spk_embedding import SpeakerEmbedding
model = SpeakerEmbedding("espnet/speaker_embedding_model")
audio, sr = librosa.load("problem_user.wav", sr=16000)
emb = model.embed(torch.tensor(audio))
```

수행 계획 및 방법

- 2단계: 화자 인식 모델 구현
 - 화자 임베딩 간 유사도 계산 알고리즘 구현
 - 임계값 설정 및 동일 화자 판별 로직 구현

```
  
# 화자 유사도 계산 및 판별 간략 코드  
def compare_speakers(emb1, emb2, threshold=0.7):  
    similarity = cosine_similarity(emb1.reshape(1, -1), emb2.reshape(1, -1))[0][0]  
    return similarity > threshold, similarity
```

기술 선택 이유

- 기존에는 ECAPA-TDNN을 사용하려 했으나
- ESPnet
 - 거대 오픈소스 (star 8.9k) 프로젝트로 접근성이 높고 커스터마이징 용이
 - 다양한 음향 환경에서의 화자 인식 성능이 검증됨 (Kaldi 기반)
 - ASR (Automatic Speaker Recognition)외에도 여러가지 부가 기능을 제공, 추후 확장성이 좋음
 - E.g.) TTS, Speech Enhancement, Speech Translation, Machine Translation, Voice Conversion...



기술 선택 이유

- 코사인 유사도
 - 화자 임베딩 벡터 간 유사도 측정에 대중적으로 사용됨
 - 계산 복잡도가 낮아 추후 실시간 시스템 확장 시 적합
 - 0-1 사이의 정규화된 값으로 임계값 설정 직관적임
- 익명 평가 시스템
 - 커뮤니티 자정 작용을 통한 문제 사용자 초기 식별 가능
 - 무분별한 음성 수집 방지 및 프라이버시 침해 최소화
 - 평가가 낮은 사용자에게 대해서만 선별적 모니터링을 함으로 전체적인 시스템 부하 감소

기대 Output

- ESPnet 기반 익명 화자 인식 모듈
- 익명 평가 및 화자 인식 연동 프로토타입

추후 R&D 방안

- VRChat, Roblox와 같은 메타버스 플랫폼에 모듈형 플러그인, 애드온으로 제공
- 메타버스 내 안전한 커뮤니티 형성을 위한 보조 도구로 활용
- 향후 메타버스 플랫폼의 사용자 보호 정책 수립에 기여

GitHub Repository

- <https://github.com/RieLCho/ESPNet-PoC>

```
@torch.cuda.amp.autocast(enabled=False)
Fetching 30 files: 100%|
/home/yangjin/ESPNet-PoC/.venv/lib/pyth
orch.load` with `weights_only=False` (
ssible to construct malicious pickle d
ch/pytorch/blob/main/SECURITY.md#untru
ly` will be flipped to `True`. This li
o longer be allowed to be loaded via th
add_safe_globals`. We recommend you st
the loaded file. Please open an issue
torch.load(model_file, map_location=
음성 파일 분석 중: ./data/mika/2.wav
/home/yangjin/ESPNet-PoC/.venv/lib/pyth
.autocast(args...)` is deprecated. Ple
with autocast(False):
알 수 없는 화자 (유사도: -1.0000)
```

```
48
49 def identify_speaker(self, audio_path, threshold=0.7):
50     """
51     입력된 음성의 화자 식별
52     Args:
53         audio_path (str): 식별할 음성 파일 경로
54         threshold (float): 유사도 임계값
55     Returns:
56         tuple: (가장 유사한 화자 ID, 유사도 점수)
57     """
58     test_embedding = self.extract_speaker_embedding(audio_path)
59
60     max_similarity = -1
61     best_speaker_id = None
62
63     for speaker_id, stored_embedding in self.speaker_embeddings.items():
64         similarity = cosine_similarity(
65             test_embedding.reshape(1, -1),
66             stored_embedding.reshape(1, -1)
67         )[0][0]
68
69         if similarity > max_similarity:
70             max_similarity = similarity
71             best_speaker_id = speaker_id
72
73     if max_similarity < threshold:
74         return None, max_similarity
75     # Ctrl+I to chat, Ctrl+K to generate
76     return best_speaker_id, max_similarity
```

참고문헌

- 송혜진, 남완우 (2022) 메타버스 내 범죄발생 유형과 확장성에 관한 연구, 한국재난정보학회논문집, 18:1, 218-227
- [arXiv:1804.00015](#) ESPnet: End-to-End Speech Processing Toolkit
- [arXiv:2401.17230](#) ESPnet-SPK: full pipeline speaker embedding toolkit with reproducible recipes, self-supervised front-ends, and off-the-shelf models



END OF DOCUMENT