SMART UP

길진성 김태성 김현욱 정선진

주간 활동 보고서

목차

<주간 활동 보고서 12 주차>
-2024.05.20(월) ~ 2024.05.26(일)-

1. 활동시간 및 내용

<스마트업 회의 및 활동 진행 시간>

-오프라인 회의 및 온라인 회의 시간
5 월 21 일(화) / 15 시~19 시 (4 시간) 오프라인
5 월 22 일(수) / 12 ~18 시 (6 시간) 오프라인
5 월 22 일(수) / 22 시~23 시 (1 시간) 디스코드 온라인
5 월 24 일(금) / 13 시~16 시 (3 시간) 오프라인
총 : 14 시간

-회의 내용

>> [스마트업] 12 주차 회의 및 활동 내용

-요약알고리즘팀 작성 코드를 라즈베리파이 환경에서 작동 가능한지 테스트를 진행해보기로함.

-평가단 모집에 대한 회의 진행
디자인과 + 컴퓨터공학과 : 5 명
생명공학과 : 5 명
평가단 모집해서 평가 진행 예정

-요약알고리즘팀 작성 코드를 긴 글에서도 문제 없이 요약이 잘 되는지 테스트해보기로 함.
1500 자씩 나눠서 요약하는 방식

-마이크 세팅 및 테스트를 진행

-주문한 물품 배송오면 조립 후 평가 진행하기로 함.

-24 일 (금) 오프라인 중간발표를 위한
PPT 내용을 작성

-12 주차 주간보고서 내용을 작성

2. 12주차 진행사항

- 12 주차 진행사항에 대한 팀 별 정리 내용

<STT 팀>

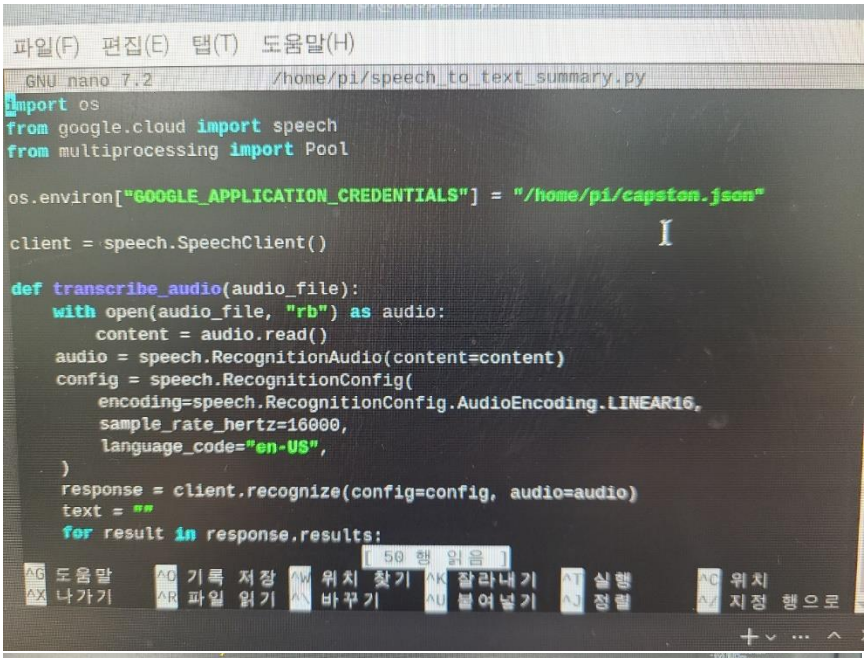

```
sudo apt update
sudo apt upgrade
aplay -l
sudo nano /etc/asound.conf
pcm.!default {
     0
    card 1
}

ctl.!default {
     0

    card 1
}
sudo alsa force-reload
speaker-test -t wav -c 2
alsamixer
sudo apt install pulseaudio
sudo apt install pavucontrol
pavucontrol
```

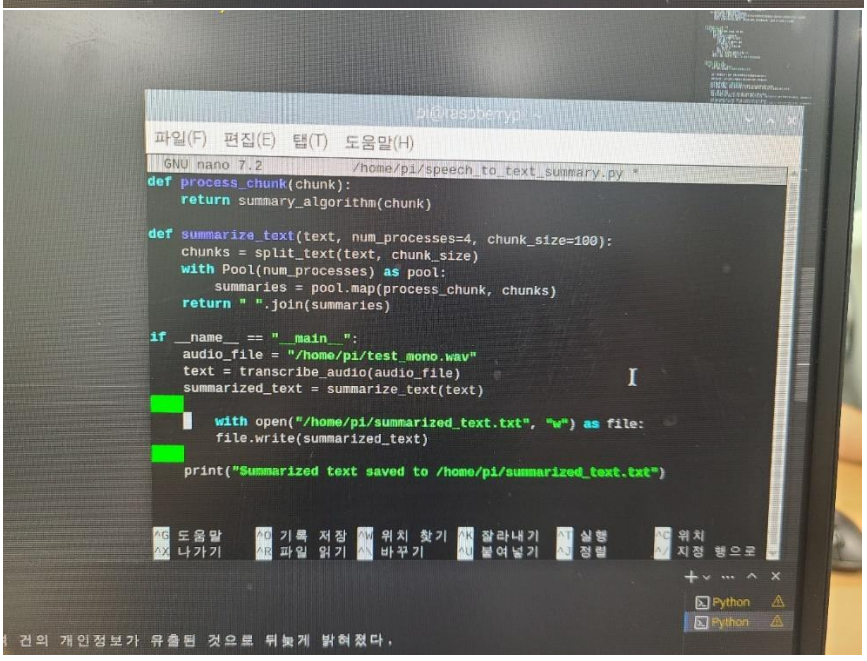
마이크와 사운드카드를 세팅하고 작동 확인을 했습니다.

```
mkdir ~/temp_gpg
curl -fsSL https://packages.cloud.google.com/apt/doc/apt-key.gpg | gpg --dearmor -o
~/temp_gpg/cloud.google.gpg
sudo cp ~/temp_gpg/cloud.google.gpg /usr/share/keyrings/
```

```
sudo apt update
sudo apt install -y ca-certificates
sudo update-ca-certificates
sudo apt install -y ntp
sudo service ntp start
sudo timedatectl set-ntp true
curl --cacert /etc/ssl/certs/ca-certificates.crt -fsSL
https://packages.cloud.google.com/apt/doc/apt-key.gpg | gpg --dearmor -o
/usr/share/keyrings/cloud.google.gpg
sudo apt update
sudo apt upgrade -y
sudo apt install -y curl apt-transport-https ca-certificates gnupg
sudo timedatectl set-ntp true
curl --cacert /etc/ssl/certs/ca-certificates.crt -fsSL
https://packages.cloud.google.com/apt/doc/apt-key.gpg | gpg --dearmor -o
/usr/share/keyrings/cloud.google.gpg
echo "deb [signed-by=/usr/share/keyrings/cloud.google.gpg]
https://packages.cloud.google.com/apt cloud-sdk main" | sudo tee
/etc/apt/sources.list.d/google-cloud-sdk.list
sudo apt update
sudo apt install -y google-cloud-sdk
pip3 install --upgrade google-api-python-client google-auth-httplib2 google-auth-
oauthlib
```

Google Assistance API 설치를 위해 초기 인증 수동 설정을 진행하고 구글 API 설치 후, 파이썬으로 API 를 정상적으로 불러왔습니다.





나노 콘솔에서 stt 변환 코드를 구현한 과정입니다

```python
def split_text_to_stack(transcript, chunk_size=1000):
    words = transcript.split()
    stack = []

    for i in range(0, len(words), chunk_size):
        chunk = " ".join(words[i:i + chunk_size])
        stack.append(chunk)

    return stack
```

STT 로 변환된 텍스트를 일정한 크기로 분할하여 스택에 넣는 코드입니다.

```python
import multiprocessing as mp

def summarize_text(text_chunk):
    # 요약알고리즘#
sentences = text_chunk.split(". ")
    if len(sentences) > 1:
        return sentences[0] + "... " + sentences[-1]
    return text_chunk

def parallel_summarize(transcript, num_processes=4):
    stack = split_text_to_stack(transcript)

    with mp.Pool(num_processes) as pool:
        summaries = pool.map(summarize_text, stack)

    return " ".join(summaries)
```

파이썬으로 Multiprocessing 을 구현하는 코드입니다.

<요약알고리즘팀>

대통령실이 13일 일본 정부의 행정지도로 촉발된 이른바 '라인 사태'와 관련해 "우리 기업의 의사에 조금이라도 반하는 부당한 조치에 단호하고 강력하게 대응할 것"이라고 밝혔다. 기업이 해외에서 불리한 처분이나 여건에 처하지 않고 자율적 의사결정을 하도록 최대한 지원하겠다는 것이다.
앞서 과학기술정보통신부도 지난 10일 비슷한 입장을 밝힌 바 있다. 이런 대응 방향과 우리 기업 지원은 정부의 당연한 책무이기도 하다. 하지만 상황은 이미 복잡한 국면으로 빠져들고 있어 입장 발표 타이밍이 너무 늦은 게 아닌가 판단된다. 일본 측 경영진의 '네이버 지우기' 행보는 현재 진행형이고, "기술 탈취" "기업 강탈"이라는 우리 국민의 반감도 커졌다.
야당은 "대일 굴종 외교"라고 가세한 형국이다.일본 정부는 개인정보 유출로 인한 행정지도에 '지분매각'이라는 직접적인 용어는 없다고 강조하지만, '네이버가 50% 출자하고 있는 자본 관계의 재검토'라는 표현은 외국 기업인 네이버의 라인 지분을 소프트뱅크에 넘기라는 압박으로 받아들일 여지가 있다.

네이버와 소프트뱅크는 라인 모회사인 A홀딩스 지분을 50%씩 보유하고 있다. 경영권 관점에서 접근한 게 아니라는 일본 정부의 해명에도, 겉으로 드러나기엔 민간 기업의 행보는 정부의 조치에 발을 맞추고 있다. 라인야후는 유일한 한국인을 이사회 멤버에서 제외하고 기술적 협력 관계인 네이버로부터 독립을 추진하겠다고 밝히는 등 '탈(脫) 네이버' 전략을 공식화했다. 소프트뱅크도 네이버와 지분 문제를 놓고 협상 중이라고 확인했다. 물론 지분 협상이 행정지도 이후 시작됐는지 그 이전에 시작됐는지는 확인되지 않았다.
이번 사태에 여론이 민감하게 반응하는 이유는 우리 기업의 원천기술이 적용되고 13년간 키워온 글로벌 플랫폼을, 적대국도 아닌 우호국이 거저먹으려 한다는 인식 때문이다. 세계 각국은 반도체, IT 등 첨단 기술 분야에서 사활을 건 패권 경쟁을 벌이고 있다. 국민은 이 사안도 단순한 기업의 경영권 문제가 아닌 경제 안보 이슈로 보는 것이다. 정부가 지분 매각 문제로만 접근해 안일하게 대응하는 게 아니냐는 지적이 나오는 이유다.
정부는 일본 측이 유사한 사례에서 다른 국내외 기업에 취한 조치와 달리 우리 기업이 차별이나 부당한 대우를 받지 않았는지 세밀하게 따져봐야 할 것이다. 상대 국가 및 해당 기업과 긴밀하게 소통하면서 국익이 침해되지 않고 기업 이익이 극대화되도록 문제를 풀어가야 할 것이다.야당이 이 상황을 정치 쟁점화해 반일 프레임으로 접근하는 점도 경계한다. 중장기적 비전이나 전략을 짜야 하고 노사 관계까지 얽힌 기업에 걸림돌이 될 수 있다.

네이버는 며칠 전 낸 입장 자료에서 "모든 가능성을 열고 소프트뱅크와 협의하고 있다"고 밝혔다. 미래성장 가능성을 높이기 위해 회사 자원의 활용과 투자에 대한 고민을 지속적으로 하고 있고, 지분 매각도 하나의 카드가 될 수 있다는 것이다. 정부 대응이 적절했는지 등은 따져 물어야 하겠지만, 국민 정서에 기대어 지나치게 기업 문제에 간섭하면 경영적 결단과 같은 활동에 지장을 줄 소지가 크다.

```
PS C:\Users\jin\.vscode> & C:/Users/jin/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/jin/.vscode/textrank&tf-idf.py"
일본 정부는 개인정보 유출로 인한 행정지도에 ' 지분 매각' 이라는 직접적인 용어는 없다고 강조하지만, ' 네이버가 50% 출자하고 있는 자본 관
들일 여지가 있다.
경영권 관점에서 접근한 게 아니라는 일본 정부의 해명에도, 겉으로 드러 나기엔 민간 기업의 행보는 정부의 조치에 발을 맞추고 있다.
소프트 뱅크도 네이버와 지분 문제를 놓고 협상 중이라고 확인했다.
['기업', '일본', '네이버', '정부', '기술', '라인', '관계', '행정지도', '경영', '매각']
```

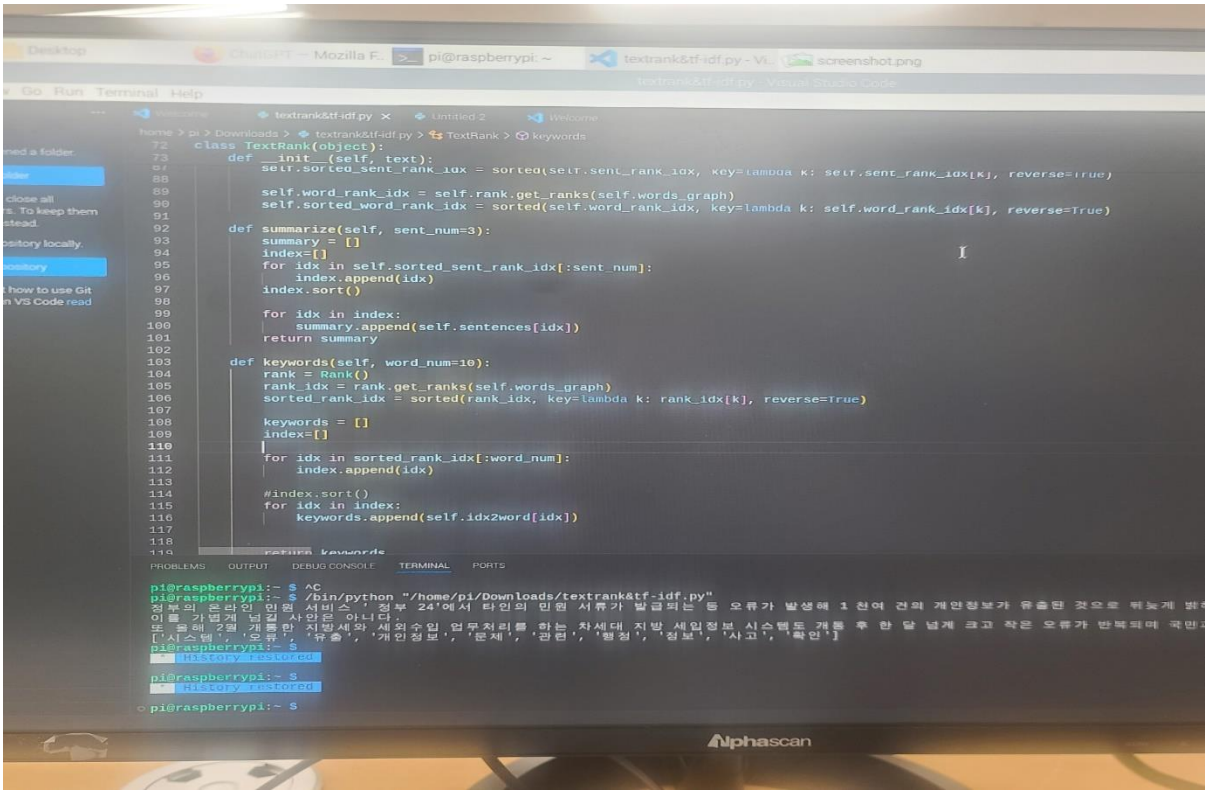
법원 전산망에 북한 해킹조직 '라자루스'로 추정되는 집단이 침투해 2년 넘게 개인정보 등이 포함된 1천GB(기가바이트) 이상의 자료를 빼낸 사실이 드러났다. 경찰청 국가수사본부가 공개한 국가정보원, 검찰과의 합동 조사 결과, 법원 전산망에 대한 침입은 2021년 1월 7일 이전부터 2023년 2월 9일까지 이뤄졌으며, 이 기간에 총 1천14GB의 법원 자료가 8대의 서버(국내 4대·해외 4대)를 통해 법원 전산망 외부로 전송됐다.
유출 자료 중 내용이 확인 가능한 것은 4.7GB 분량인 파일 5천171개로 전체의 0.5%에 불과하다고 한다. 적어도 2년 이상 사법부 전산망이 해킹에 지속 노출됐고 유출된 자료 규모가 방대하다는 점은 충격적이다.법원 전산망에는 일반 시민은 물론 국내외 기업과 수사기관, 정부 부처, 금융당국 등이 제출한 유출 시 악용될 우려가 큰 수많은 민감한 정보가 모여있다.

이번에 유출 확인된 자료 5천171개만 하더라도 자필진술서, 채무증대 및 지급불능 경위서, 혼인관계증명서, 진단서 등으로, 여기에는 이름, 주민등록번호, 금융정보, 병력기록 등 개인정보가 다수 포함됐다.
그런데 이번에 내용이 확인되지 못한 나머지 유출 자료 99.5%에는 또 어떤 민감한 정보가 담겨 있을지조차 모른다. 법원의 사후 '늑장 대응' 논란도 제기된다. 법원 내부망에서 백신이 악성코드를 감지해 차단한 시점은 작년 2월 9일이지만, 대법원이 자체 대응하면서 경찰 수사는 언론 보도로 해킹 사건이 처음 알려진 뒤인 작년 12월 5일에야 시작됐다.

해킹 범행이 일어난 지 한참 뒤에 수사가 이뤄지면서 서버에 남아있던 유출 자료들이 지워졌고 해킹 경로나 목적도 확인하지 못했다. 법원의 해킹 차단 보안시스템도, 사후 대응 조치도 문제였던 셈이다.사안의 엄중함에 걸맞은 재발 방지 시스템 강화가 필요하다. 전문 인력과 장비 확충 등을 통해 보안시스템 취약점은 보완하고 대응 체제를 개편해야 한다.
사법부의 독립성을 저해하지 않으면서도 사이버 보안과 관련해 관련 기관과의 협력, 공조 강화 방안도 모색해야 한다. 해킹에 대한 예방과 사후 대응이 부실했다면 그 경위도 조사하고 엄중히 책임을 물어야 할 것이다. 수사당국은 범행에 사용된 악성 프로그램 유형, 가상자산을 이용한 임대서버 결제내역, IP 주소 등을 바탕으로 이번 사건을 북한 해킹조직의 소행으로 결론 내렸다.
북한은 최근 민간, 공공부문을 가리지 않고 우리를 겨냥한 해킹 시도를 증가하고 있다. 올해 초 국정원 자료에 따르면 지난해 국내 공공분야를 대상으로 하루 평균 162만여건의 국가 배후 및 국제 해킹조직의 공격 시도를 탐지했으며, 공격 주체별로는 북한이 80%로 가장 많았다.
초유의 사법부 전산망 해킹을 계기로 경각심을 다시 한번 높여야 한다.

```
PS C:\Users\jin\.vscode> & C:/Users/jin/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/jin/.vscode/textrank&tf-idf.py"
법원 전산망에 북한 해킹조직 ' 라자 루스' 로 추정되는 집단이 침투해 2년 넘게 개인정보 등이 포함된 1천 GB( 기가 바이트) 이상의 자료를 빼낸
해킹 범행이 일어난 지 한참 뒤에 수사가 이뤄 지면서 서버에 남아 있던 유출 자료들이 지워 졌고 해킹 경로나 목적도 확인하지 못했다.
법원의 해킹 차단 보안 시스템도, 사후 대응 조치도 문제였던 셈이다.
['자료', '해킹', '법원', '전산', '유출', '북한', '서버', '조직', '국가', '국내']
```

작성한 코드가 긴 글에서도 문제없이 요약되는지 테스트했습니다.



개발한 요약알고리즘 코드가 라즈베리파이 환경에서도 문제없이 작동하는지 실제 적용해봤습니다.

```python
import multiprocessing
import time
from queue import Queue

# 샘플 텍스트 데이터를 생성하는 함수 (실제 STT 대신 사용)
def generate_sample_text():
    return "이것은 샘플 텍스트입니다. " * 100

# Process 1: STT 기능을 시뮬레이션하여 텍스트를 생성하고 Queue에 넣기
def stt_process(text_queue):
    buffer = ""
    while True:
        text = generate_sample_text()
        buffer += text
        if len(buffer) >= 1500:
            text_queue.put(buffer)
            buffer = ""
        time.sleep(1)  # 실제 구현에서는 STT 모듈을 통해 음성을 지속적으로 받아야 함

# Process 2: Queue에서 텍스트를 받아 요약
def summary_process(text_queue):
    while True:
        if not text_queue.empty():
            text = text_queue.get()
            summary = summarize_text(text)
            print(f"Summary: {summary}")
        time.sleep(1)
```

멀티프로세싱을 통해 작업시간을 단축하고자 했고 STT 를 통해 텍스트를 받아들이는 부분과 요약알고리즘을 실행하는 부분을 두 개의 프로세스로 분화해 작업했습니다.

1 번 프로세스는 STT 를 통해 받아들인 텍스트를 버퍼에 저장 후 , 버퍼의 크기가 1500 자가 넘으면 버퍼를 요약알고리즘을 실행하는 2 번 프로세스로 전달합니다. 1 번 프로세스는 이를 반복하고 전달받은 2 번 프로세스는 요약알고리즘을 통해 요약합니다.

```
28
29  # 간단한 텍스트 요약 함수 (여기서는 간단히 텍스트의 앞부분을 반환)
30  def summarize_text(text):
31      # sumy 등 실제 요약 알고리즘을 사용할 수 있습니다.
32      return text[:500] + "..."
33
34  if __name__ == '__main__':
35      # 텍스트를 전달할 Queue 생성
36      text_queue = multiprocessing.Queue()
37
38      # 두 개의 프로세스 생성
39      p1 = multiprocessing.Process(target=stt_process, args=(text_queue,))
40      p2 = multiprocessing.Process(target=summary_process, args=(text_queue,))
41
42      # 프로세스 시작
43      p1.start()
44      p2.start()
45
46      # 프로세스가 종료될 때까지 대기
47      p1.join()
48      p2.join()
49
```

코드가 실행되고 꺼지게 끔 활성화 시킨 코드입니다.

3. 추후 일정

- **앞으로 남은 기간 동안의 일정입니다.**

[13 주차]

- **물품배송 오면 조립을 진행할 계획입니다.**
- **다른 계열 학과 수업 3건에 대해 각각 30분 이상 시스템을 작동시키고 10명 이상의 별도 평가단을 구성하여 요약 내용 적절성에 대한 평가를 진행할 계획입니다.**

[14 주차]

- **평가 결과를 바탕으로 부족한 부분을 보완할 계획입니다.**

[15 주차]

- **캡스턴디자인 최종 발표**