

주간 활동 보고서

목차

1. 활동시간 및 내용

- 정기회의 진행 시간 및 내용
- 디스코드 온라인 회의 내용

2. 11주차 진행사항

- STT팀
- 요약알고리즘팀

3. 추가 구매 내역

- 사비 구매 물품 허가
- 추가 구매 요청

4. 다음주(12주차) 일정

- 12주차 일정

<주간 활동 보고서 11 주차> -2024.05.13(월) ~ 2024.05.19(일)-

1. 활동시간 및 내용

<스마트업 정기회의 진행 시간>

-정기 회의 및 온라인 회의 시간

5 월 14 일(화) / 18 시~20 시 30 분 (2 시간 30 분) 오프라인

5 월 16 일(목) / 19 시 40 분~20 시 (20 분) 디스코드 온라인

-회의 내용

>> [스마트업] 5 월 14 일 정기회의 내용

-STT 팀 사운드드라이버 설치문제에 대한

해결방안 논의. 이번주에 최대한 해결 시도

>> 해결 불가능 판단될 경우에는 다른 모듈
사비로 주문해서 진행하기로 함.

-이번주 제출할 영상에 대한 회의

이번주 중간점검은 영상 제출. 영상 촬영방법과
일정에 대한 회의 진행. 이번주차는 STT 팀과
요약알고리즘팀 영상 따로 제작 진행.

-이번주 주간보고서 작성에 대한 회의
이번주 주간보고서 작성은 STT 팀,알고리즘팀
따로 작성해서 취합하기로 함.

-물품 추가 구매에 대한 회의
강의 내용 요약 결과에 대한 평가단을 구성하여
평가를 진행하기 위해 라즈베리파이용
터치스크린과 케이스를 추가구매요청을 통해
구매하기로 함. 요약된 강의 내용을 직접 보고
평가 진행.

-보고서 작성 및 ppt 제작을 위해 개발중인
내용에 대한 사진첨부 논의

-요약알고리즘 평가를 위한 데이터셋 조사
Ai hub 링크

[https://www.aihub.or.kr/aihubdata/data/view.do
?currMenu=&topMenu=&aihubDataSe=data&data
SetSn=582](https://www.aihub.or.kr/aihubdata/data/view.do?currMenu=&topMenu=&aihubDataSe=data&dataSetSn=582)

>> [스마트업] 5 월 16 일 디스코드 온라인 회의 내용

1. 기존에는 정기회의를 화,수로 정함. 12주차는 진행할 일들이 많은 관계로 정기회의를 따로 정하지 않고 팀원들이 모일 수 있는 시간대에는 최대한 모여서 진행하기로 함.

2. 사운드 드라이버 설치를 라즈베리파이 말고 다른 노트북,컴퓨터에서 설치 시도해보기로함.

>> respeaker 2 pi hat 모듈의 사운드드라이버는 라즈베리파이 전용이라 노트북에 설치해도 사운드드라이버 문제 원인을 파악할 수 없다는 판단. 설치 시도 X

3. 사운드를 위한 부품을 사비로 빠르게 구매하기 위해 구매방식을 논의함.

4. 과방에 있는 물품들과 비용문제에 대한 회의

>>모니터,파워선,SD카드,멀티탭 등 현재까진 비용문제 발생X
새로운 모듈 구매는 구매 후 n분의1로 비용 지불하기로 함.

5. 터치스크린/케이스 구매에 대한 공간 문제

>>정기회의에서 구매하기로 했던 터치스크린과 케이스를 연결하기 위해선 라즈베리파이의 GPIO에 선을 연결해야 하는데

Respeaker 2 pi hat 모듈이 라즈베리파이의 GPIO공간을 모두 차지함. 터치스크린,케이스 구매를 변경할지에 대한 논의 진행.

추가구매물품

7인치 스크린 <https://www.devicemart.co.kr/goods/view?no=1273487>

7인치 스크린용 케이스

<https://www.devicemart.co.kr/goods/view?no=15348490> (수정됨)

디바이스마트

라즈베리파이 공식 7인치 터치스크린 (Raspberry Pi Touch Display)

Raspberry Pi 재단의 공식 Touchscreen Display 제품 / 해상도 800 x 480 / 라즈베리파이4.5 라즈베리파이3B+, 3B 호환



디바이스마트

라즈베리파이5 7인치 터치스크린용 메탈 케이스

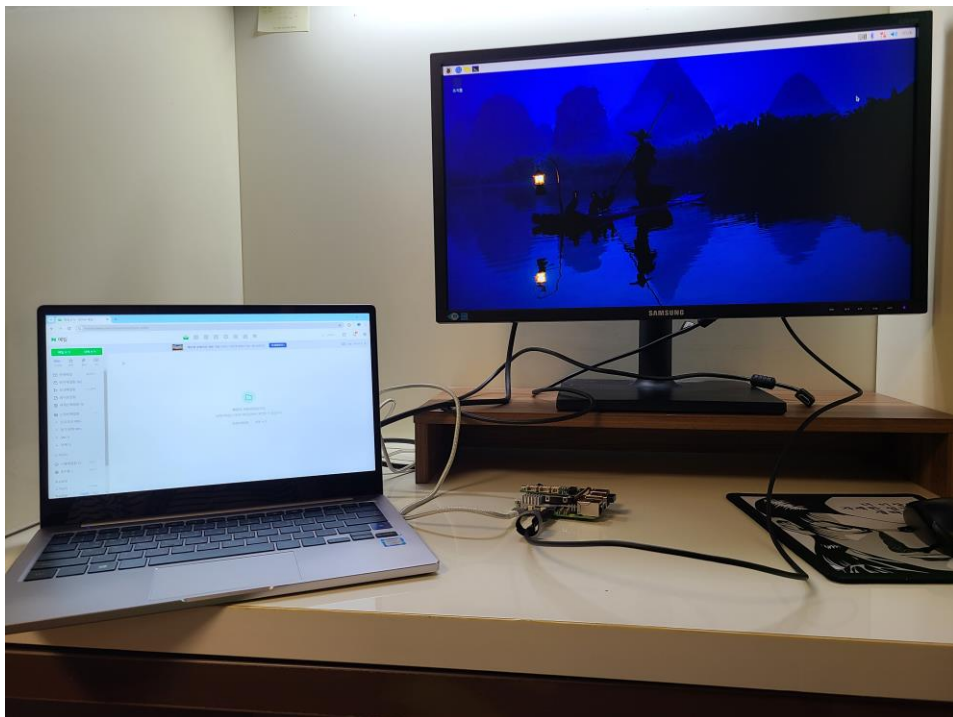
한국공식대리점 / 라즈베리파이5 호환 7인치 터치스크린용 케이스 / 액티브 쿨러 장착 가능 / 완벽호환 / 깔끔한 마감 / 구성품 : 케이스 x1, 디스플레이 케이블 x1, 핀셋 x1, 기타 피스 / Case for Raspberry Pi 5 and the Official Raspberry Pi 7-inch Touchscreen



2. 11주차 진행사항

- 11 주차 진행사항에 대한 팀 별 정리 내용

<STT 팀>



집에서도 라즈베리파이를 이용하여 여러 번 시도하는 등
지속적으로 사운드 드라이버를 설치 시도했으나,
계속되는 설치 오류로 sd 카드 초기화를 여러 번 반복했습니다.
이대로는 진도에 영향이 클 것이라 판단하여 사운드 드라이버를
제외한 STT 코딩 부분을 선행적으로 진행했습니다.

개발에 사용되는 모든 프로그램은 os 와의 충돌을 방지하기 위해
가상환경에 설치하도록 했습니다. 또한, 별도의 부트로더를 라즈베리
파이가 설치되어 있는 SD 카드에 설치했습니다.

추후에 혹시 모를 초기화 이슈가 다시 발생할 수 있기 때문에 빠른
라즈베리파이 환경 세팅을 위해
라즈베리파이 세팅과정에 대한 내용을 정리했습니다.

// 다음은 라즈베리파이 세팅 과정에 대한 정리 내용입니다.

필수) 라즈베리파이에서 원가 선을 빼거나 꽂을 때는 전원을 꺼놓은
상태에서 진행해야 함.

라즈비안 os 초기화 후 진행 순서를 작성합니다. >> 라즈베리파이
메인화면 나오기 전에 sd 카드 연결하자마자 뜨는 세팅 부분

<Set country>

Country : United Kingdom

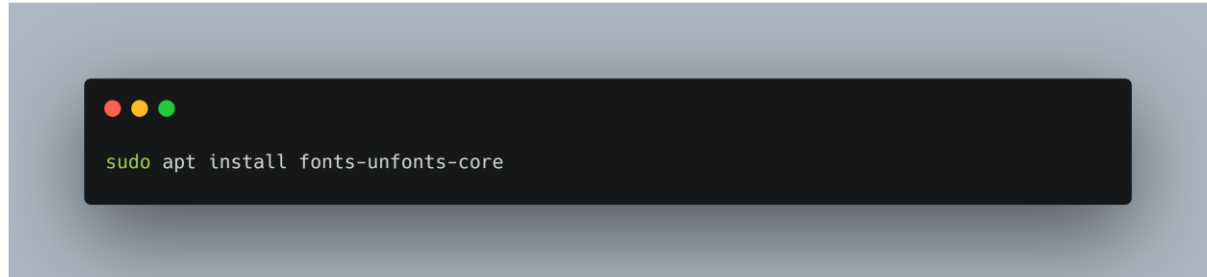
Language : British English

Timezone : London

라즈베리파이 이름은 pi 로 설정.

진행 순서

- 라즈베리파이 초기 설정 진행 상황을 작성합니다.



>한글 폰트 설치

라즈베리파이 아이콘(시작) – preferences – RPI Configuration 선택
Localisation 탭 이동

<Set Locale>

Language : ko(Korean)

Character Set : UTF-8 <OK>

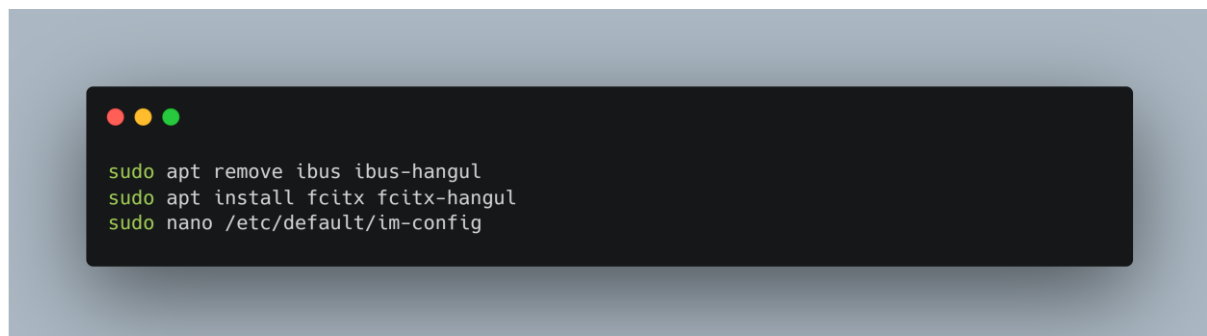
<Set Timezone>

Area : Asia

Location : Seoul <OK>

<OK> – <Yes> 선택

>언어 한글로 변환



Config 창에서

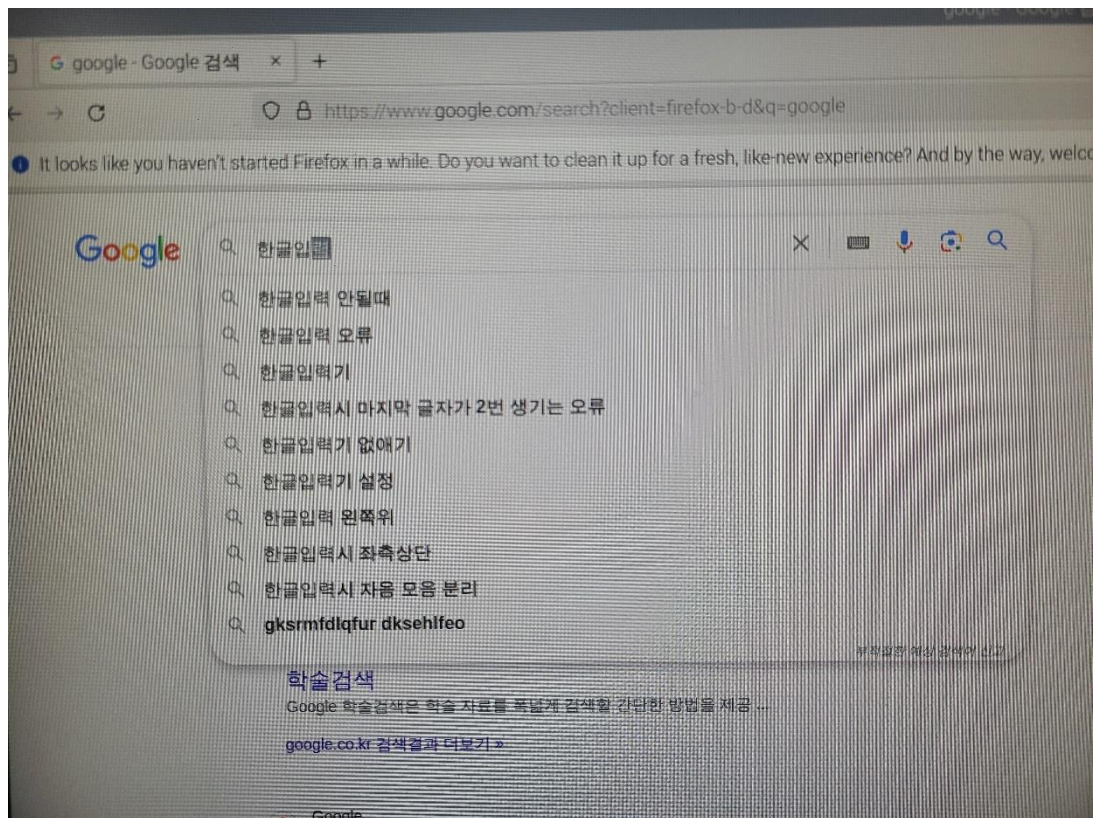
IM_CONFIG_DEFAULT_MODE=auto

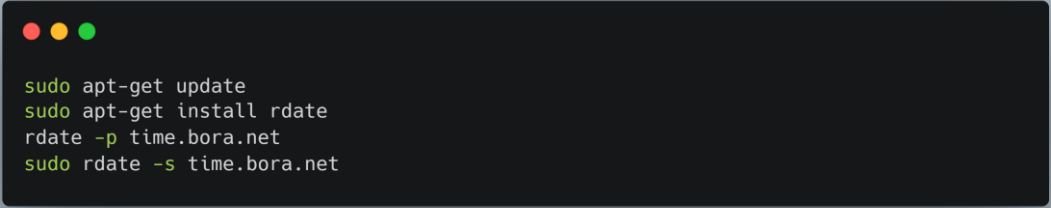
이 부분에서 auto 를 지우고 fcitx 로 변경.

<ctrl + x> - <shift + y> - <Enter>

라즈베리파이 재부팅

>한글 설정완료 , 키보드 한/영 키로 한글입력 가능.





```
sudo apt-get update
sudo apt-get install rdate
rdate -p time.bora.net
sudo rdate -s time.bora.net
```

>현재 시간 조회 및 시간 동기화

>>시간동기화를 진행하면 인터넷 접속됨.

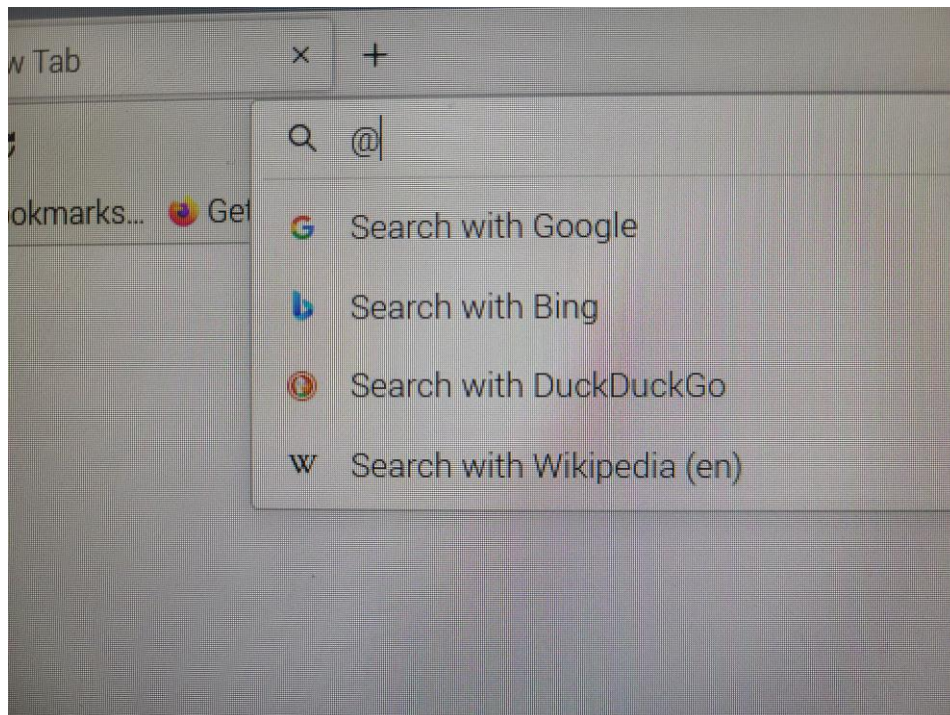
라즈베리파이 아이콘(시작) – preferences – RPI Configuration 선택
Localisation 탭 이동

<Set Keyboard>

Layout : US

Variant : US

>특수문자 키보드와 다르게 이상하게 출력되는 부분 해결.



원격접속 : SSH 활성화를 진행해야 putty 활성화 가능
SSH 활성화 코드

```
sudo service ssh status
sudo service ssh start
sudo raspi-config
sudo reboot
```

순서대로 진행하다가 sudo raspi-config 입력하면 config 창이 뜬.
<메뉴에서 interface options 선택 , SSH 선택, Would you like the SSH server to be enabled? 질문에 yes 선택 , 설정 완료 후 이미지 내용대로 sudo reboot 사용해서 재부팅>

이후, putty 활성화 진행.

// 다음은 STT 코딩 부분 진행사항입니다.
>부트로더 별도 설치**

```
sudo mkdir -p /mnt/boot
sudo mount /dev/sdX1 /mnt/boot
sudo cp recovery.bin /mnt/boot
sudo umount /mnt/boot
vcgencmd bootloader_version
```

>가상환경 설정**

```
python3 -m venv myenv
source myenv/bin/activate
```

끝나고 deactivate

>GOOGLE ASSISTANCE API 설치**

초기**인증수동설정

```
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/
```

설치 순서

```
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-http2 google-auth-
oauthlib
```


-구글 API 설치 후 , 파이썬으로 API 를 정상적으로 불러옴.


```
pi@raspberrypi: ~/seeed-voicecard
File Edit Tabs Help

[48.7 kB]
Fetched 48.7 kB in 0s (411 kB/s)
Selecting previously unselected package dkms.
(Reading database ... 208372 files and directories currently installed)
Preparing to unpack .../dkms_3.0.10-8+deb12u1_all.deb ...
Unpacking dkms (3.0.10-8+deb12u1) ...
Setting up dkms (3.0.10-8+deb12u1) ...
Processing triggers for man-db (2.11.2-2) ...
grep: /boot/kernel.img: No such file or directory

gzip: stdin: unexpected end of file
grep: /boot/kernel.img: No such file or directory

gzip: stdin: unexpected end of file
-----
WARNING Your loaded kernel version is 6.6.20
Not matching the updated version .
Kernel was updated, but new kernel was not loaded yet
Please reboot your machine AND THEN run this script again
(myenv) pi@raspberrypi:~/seeed-voicecard $ arecord -l
bash: arecord: command not found
(myenv) pi@raspberrypi:~/seeed-voicecard $ arecord -l
**** List of CAPTURE Hardware Devices ****
(myenv) pi@raspberrypi:~/seeed-voicecard $
```



여기까지 진행한 이후, 마지막에 SD 카드에 사운드 드라이버를 한번 더 재설치 시도해봤지만 다시 오류가 발생해 respeaker 2 pi hat 모듈을 사용하지 않기로 하고 사운드카드와 마이크를 새로 구매해서 빠르게 다시 세팅을 진행할 계획입니다.

<요약알고리즘팀>

저희가 작성한 알고리즘은 텍스트 랭크 알고리즘과 TF-IDF 를 결합한 알고리즘으로 대학 강의를 요약하고자 했습니다.

코드를 보면서 코드에 대한 간략한 설명을 하고 코드를 작동 후 비교를 위해 일반 텍스트 랭크 알고리즘에서 같은 문서를 작동한 결과를 비교해보겠습니다.

```
10 # 문장 추출
11 class SentenceTokenizer(object):
12     def __init__(self):
13         self.kkma = Kkma()
14         self.okt = Okt()
15         self.stopwords = ["아", "휴", "아이구", "아이쿠", "아이고", "어", "나", "우리", "저희", "따라", "의해",
16                           "을", "를", "에", "의", "가", "중간", "만큼", "마침까지"]
17
18
19     def text2sentences(self, text):
20         sentences = self.kkma.sentences(text)
21         for idx in range(0, len(sentences)):
22             if len(sentences[idx]) <= 10:
23                 sentences[idx-1] += (' ' + sentences[idx])
24                 sentences[idx] = ''
25         return sentences
26
27     def get_nouns(self, sentences):
28         nouns = []
29         for sentence in sentences:
30             if sentence != '':
31                 nouns.append(' '.join([noun for noun in self.okt.nouns(str(sentence))
32                                         if noun not in self.stopwords and len(noun) > 1]))
33         return nouns
34
35
```

먼저 문장 추출 부분에서 클래스 SentenceTokenizer 가 정의되어 있습니다. 이 클래스는 문장을 추출하고 명사를 추출하는 데 사용됩니다.

그리고 Kkma 와 Okt 형태소 분석기를 사용하여 입력된 텍스트를 문장으로 분리하고, 명사를 추출합니다. 또한, 10 자 이하의 문장은 그 다음 문장과 합쳐져 처리됩니다. 이를 통해 텍스트에서 의미 있는 문장과 명사를 추출하게 됩니다.

```
36
37 # TF-IDF 모델 생성 및 그래프 생성
38 class GraphMatrix(object):
39     def __init__(self):
40         self.tfidf = TfidfVectorizer()
41         self.cnt_vec = CountVectorizer()
42         self.graph_sentence = []
43     def build_sent_graph(self, sentence):
44         tfidf_mat = self.tfidf.fit_transform(sentence).toarray()
45         self.graph_sentence = np.dot(tfidf_mat, tfidf_mat.T)
46         return self.graph_sentence
47     def build_words_graph(self, sentence):
48         cnt_vec_mat = normalize(self.cnt_vec.fit_transform(sentence).toarray().astype(float), axis=0)
49         vocab = self.cnt_vec.vocabulary_
50         return np.dot(cnt_vec_mat.T, cnt_vec_mat), {vocab[word] : word for word in vocab}
51
52
```

이번에는 TF-IDF 모델을 생성하고 그래프를 생성하는 부분에 대한 코드입니다.

GraphMatrix 클래스를 정의하고, TF-IDF 모델을 사용하여 문장 그래프와 단어 그래프를 생성합니다.

build_sent_graph 함수는 주어진 문장에서 TF-IDF 행렬을 생성하고, 이를 이용하여 문장 간 유사도를 계산하여 문장 그래프를 생성합니다.

build_words_graph 함수는 주어진 문장에서 단어 빈도 행렬을 생성하고, 이를 이용하여 단어 간 유사도를 계산하여 단어 그래프를 생성합니다.

이를 통해 각 문장과 단어의 유사도를 기반으로 그래프를 생성할 수 있습니다.

```

54 #TextRank 알고리즘 적용
55 class Rank(object):
56     def get_ranks(self, graph, d=0.85):
57         A = graph
58         matrix_size = A.shape[0]
59         for id in range(matrix_size):
60             A[id, id] = 0
61             link_sum = np.sum(A[:,id])
62             if link_sum != 0:
63                 A[:, id] /= link_sum
64             A[:, id] *= -d
65             A[id, id] = 1
66         B = (1-d) * np.ones((matrix_size, 1))
67         ranks = np.linalg.solve(A, B)
68         return {idx: r[0] for idx, r in enumerate(ranks)}
69

```

이 코드는 TextRank 알고리즘을 활용하여 주어진 그래프에서 각 노드 즉 문장 또는 단어의 중요도를 계산합니다.

페이지 랭크를 계산하여서 각 노드의 중요도를 숫자로 나타내어 그래프에 대한 스코어를 계산합니다.

```

70
71 #TextRank Class 구현
72 class TextRank(object):
73     def __init__(self, text):
74         self.sent_tokenize = SentenceTokenizer()
75
76
77         self.sentences = self.sent_tokenize.text2sentences(text)
78
79         self.nouns = self.sent_tokenize.get_nouns(self.sentences)
80
81         self.graph_matrix = GraphMatrix()
82         self.sent_graph = self.graph_matrix.build_sent_graph(self.nouns)
83         self.words_graph, self.idx2word = self.graph_matrix.build_words_graph(self.nouns)
84
85         self.rank = Rank()
86         self.sent_rank_idx = self.rank.get_ranks(self.sent_graph)
87         self.sorted_sent_rank_idx = sorted(self.sent_rank_idx, key=lambda k: self.sent_rank_idx[k], reverse=True)
88
89         self.word_rank_idx = self.rank.get_ranks(self.words_graph)
90         self.sorted_word_rank_idx = sorted(self.word_rank_idx, key=lambda k: self.word_rank_idx[k], reverse=True)
91
92     def summarize(self, sent_num=3):
93         summary = []
94         index=[]
95         for idx in self.sorted_sent_rank_idx[:sent_num]:
96             index.append(idx)
97         index.sort()
98
99         for idx in index:
100             summary.append(self.sentences[idx])
101         return summary
102
103     def keywords(self, word_num=10):
104         rank = Rank()
105         rank_idx = rank.get_ranks(self.words_graph)
106         sorted_rank_idx = sorted(rank_idx, key=lambda k: rank_idx[k], reverse=True)
107

```


이 코드는 TextRank 알고리즘을 사용하여 텍스트를 요약하고, 키워드를 추출하는 클래스를 구현한 것입니다.

TextRank 클래스는 주어진 텍스트를 받아 초기화합니다.

SentenceTokenizer 를 사용하여 텍스트를 문장으로 분리하고, 명사를 추출합니다.

GraphMatrix 를 사용하여 문장 그래프와 단어 그래프를 구축합니다.

Rank 클래스를 사용하여 문장과 단어의 페이지 랭크를 계산합니다.

summarize 메서드는 페이지 랭크가 높은 순서대로 일정 개수의 문장을 선택하여 요약을 생성합니다.

keywords 메서드는 페이지 랭크가 높은 순서대로 일정 개수의 단어를 선택하여 키워드를 생성합니다.

이렇게 구현된 클래스는 텍스트 요약과 키워드 추출을 위한 인터페이스로 사용됩니다.

```
108     keywords = []
109     index=[]
110
111     for idx in sorted_rank_idx[:word_num]:
112         index.append(idx)
113
114     #index.sort()
115     for idx in index:
116         keywords.append(self.idx2word[idx])
117
118
119     return keywords
120
121
122
123     text = #요약할 텍스트를 입력
124     textrank = TextRank(text)
125     for row in textrank.summarize(3):
126         print(row)
127         print('\n')
128         print('\n')
129     print(textrank.keywords())
```

마지막으로 요약할 텍스트를 입력하고 텍스트를 몇줄로 요약할 것인지 설정합니다.

```
72 class TextRank(Object):
103     def keywords(self, word_num=10):
106         sorted_rank_idx = sorted(rank_idx, key=lambda k: rank_idx[k], reverse=True)
107
108         keywords = []
109         index=[]
110
111         for idx in sorted_rank_idx[:word_num]:
112             index.append(idx)
113
114         #index.sort()
115         for idx in index:
116             keywords.append(self.idx2word[idx])
117
118
119         return keywords
120
121
122
123 text = "대통령실이 13일 일본 정부의 행정지도로 촉발된 이른바 '라인 사태'와 관련해 '우리 기업의 의사에 조금이라도 반하는 부당한 조치에 단호하"
124 textrank = TextRank(text)
125 for row in textrank.summarize(3):
126     print(row)
127     print('\n')
128     print('\n')
129 print(textrank.keywords())
```

문제 출력 디버그 콘솔 터미널 포트

```
P5 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 관계의 재검토' 라"
들일 여지가 있다.
```

경영권 관점에서 접근한 게 아니라는 일본 정부의 해명에도, 겉으로 드러 나가면 민간 기업의 행보는 정부의 조치에 발을 맞추고 있다.

소프트뱅크도 네이버와 지분 문제를 놓고 협상 중이라고 확인했다.

['기업', '일본', '네이버', '정부', '기술', '라인', '관계', '행정지도', '결정', '매각']

```
P5 C:\Users\jin\.vscode>
```

먼저 해당 알고리즘이 정상작동하는 모습입니다

다음으로 해당 알고리즘에 5 개의 문서를 넣고 동작한 결과와 같은
문서를 일반 TEXTRANK 알고리즘에 넣은 결과를 비교해보겠습니다.

위가 개발 알고리즘 아래가 텍스트 랭크 단일 알고리즘입니다.

대학교 강의를 위한 텍스트 요약이기에 무조건 짧게 요약하는것보다는
핵심내용이 잘 남아있도록 개발하고자 합니다.

1.

```
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 '지문 매각'이라는 직접적인 용어는 없다고 강조하지만, '네이버가 56% 차지하고 있는 지문 관계의 재검토'라는 표현은 외국 기업이 네이버의 라인 지문을 소프트뱅크에 넘기라는 압박으로 받아들이는 여지가 있다.
경정권 과정에서 접근한 게 아니라는 일본 정부의 해명에도, 결국로 드러 나기엔 민간 기업의 행보는 정부의 조치에 발을 맞추고 있다.
소프트뱅크도 네이버와 지문 문제를 놓고 항상 중이라고 확인했다.
[기업, 일본, 네이버, 정부, 기술, 라인, 관계, 행정지도, 경정, 매각]
```

2.

```
PS C:\Users\jin\.vscode> & C:\Users\jin\AppData\Local\Microsoft\WindowsApps\python3.11.exe c:/Users/jin/.vscode/only_textrank.py
대통령실이 13일 일본 정부의 행정지도로 촉발한 이른바 '라인 사태'와 관련해 우리 기업의 외사에 조금이라도 반하는 부당한 조치에 단호하고 강력하게 대응할 것이라고 밝혔다. 이번 사태에 여론이 민감하게 반응하는 이유는 우리 기업의 원천기술이 적용되고 13년간 키워온 글로벌 플랫폼을, 적대국도 아닌 우호국이 거저머으려 한다는 인식 때문이다
```

3.

```
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 묵한 해킹조직 '라자 루스'로 추정되는 집단이 침투해 2년 넘게 개인정보 등이 포함된 1천 66(기가 바이트) 이상의 자료를 빼낸 사실이 드러났다.
해킹 행위가 일어난 지 한참 뒤에 수사가 이뤄지면서 서버에 남아 있던 유출 자료들이 지워졌고 해킹 경로나 목적도 확인하지 못했다.
법원의 해킹 차단 보안 시스템도, 사후 대응 조치도 문제였던 셈이다.
[지문, 해킹, 법원, 전산, 유출, 묵한, 서버, 조직, 국가, 국내]
```

```
PS C:\Users\jin\.vscode> & C:\Users\jin\AppData\Local\Microsoft\WindowsApps\python3.11.exe c:/Users/jin/.vscode/only_textrank.py
법원 내부망에서 백신이 악성코드를 감지해 차단한 시점은 작년 2월 9일이지만, 대법원이 자체 대응하면서 경찰 수사는 언론 보도로 해킹 사건이 처음 알려진 뒤인 작년 12월 5일해야 시작했다. 해킹에 대한 예방과 사후 대응이 부신했다면 그 경위도 조사하고 엄중히 책임을 물어야 할 것이다
```

4.

```
PS C:\Users\jin\.vscode> & C:\Users\jin\AppData\Local\Microsoft\WindowsApps\python3.11.exe "c:/Users/jin/.vscode/textrank&tf-idf.py"
법원에 의해 2천명 증원과 대학별 배운 법정의 타당성을 따지기 위해 정부에 각종 근거 자료를 요청하자 정부와 의료계가 회의록 존재 여부 등을 놓고 소모적인 논쟁을 벌이고 있다.
정부가 그동안 운영한 회의제 또는 협의제는 의료 현안 협의제, 보건의료정책 심의 위원회(보정 심), 보정 심 산하 의사인력전문위원회(전문 위), 전문 배경심사위원회(배정 위) 등이다.
근거야 '소송 중이라 회의록 유무를 확인할 수 없는 이유도, 입장을 바꾼 이유도 알려줄 수 없다'고 한다.
[회의, 의료, 정부, 증원, 정책, 의대, 법원, 법원, 합의, 여부]
```

```
PS C:\Users\jin\.vscode> & C:\Users\jin\AppData\Local\Microsoft\WindowsApps\python3.11.exe c:/Users/jin/.vscode/only_textrank.py
그러나 의대 증원 문제와 같은 사회적으로 중치대하고 이해관계가 첨예한 정책을 결정할 때는 모든 회의 내용을 비공개로라도 기록으로 작성해 일정 기간 보관하는 게 상식이다. 국민도 이를 믿고 정부의 의대 증원을 포함한 의료 개혁 방침에 압도적으로 찬성했다
```

5.

```
PS C:\Users\jin\.vscode> & C:\Users\jin\AppData\Local\Microsoft\WindowsApps\python3.11.exe "c:/Users/jin/.vscode/textrank&tf-idf.py"
의료 제정과 노후 소득 보장을 주요 정책을 제대로 풀어내지 못한 데 대해 여야 모두 자성해야 한다.
모두 개혁의 연금 개혁의 핵심인데 '행정' 개혁이라는 비판이 나올 수밖에 없었다.
이를 놓고 연금 개혁의 골든 타임을 이미 놓친 게 아니냐는 우려는 커졌다.
[개혁, 연금, 논의, 국회, 소득, 여야, 일기, 대세, 모수, 국민연금]
```

```
PS C:\Users\jin\.vscode> & C:\Users\jin\AppData\Local\Microsoft\WindowsApps\python3.11.exe c:/Users/jin/.vscode/only_textrank.py
국민연금 보험료율과 지급액을 조정하기 위한 연금 개혁이 2대 국회에서 무산될 무기에 처했다. 이번 국회 임기가 끝나기 전에 역시 시급한 모수 개혁이라도 마무리 지은 뒤 22대 국회에서 전반적인 구조개혁 문제를 함께 논의할 수도 있다
```

6.

```
PS C:\Users\jin\.vscode> & C:\Users\jin\AppData\Local\Microsoft\WindowsApps\python3.11.exe "c:/Users/jin/.vscode/textrank&tf-idf.py"
정부의 온라인 민원 서비스 '정부 24'에서 타인의 민원 서류가 발급되는 등 오류가 발생해 1천여 건의 개인정보가 유출된 것으로 뒤늦게 밝혀졌다.
이를 가림막에 넘긴 사안은 아니다.
또 올해 2월 개편한 지방세와 세외수입 업무처리를 하는 차세대 지방 세입정보 시스템도 개편 후 한 달 넘게 크고 작은 오류가 반복되며 국민과 공무원들에게 불편을 끼쳤다.
[시스템, 오류, 유출, 개인정보, 문제, 관련, 행정, 정보, 사고, 확인]
```

```
PS C:\Users\jin\.vscode> & C:\Users\jin\AppData\Local\Microsoft\WindowsApps\python3.11.exe c:/Users/jin/.vscode/only_textrank.py
정부 전산망과 관련된 크고 작은 오류가 이어지며 신뢰가 떨어진다면 어떻게 이를 이용하는 국민이 불안해하지 않을 수 있겠는가. 정부 행정전산망과 관련한 이런저런 사건 사고가 끊이지 않고 있는 것은 우려를 낳을만한 문제다
```

3. 추가 구매 내역

- 사운드 드라이버 설치 문제로 인해 기존 respeaker 2 pi hat 모듈 사용이 불가능하다고 판단되어 STT 기능 구현을 위한 사운드카드와 마이크를 사비로 추가 구매 진행했습니다.



필라 컴소닉 USB 외장 사운드 카드

★★★★★ 226개 상품평

4,030원 로켓배송

무료배송 (로켓배송 상품 19,800원 이상 구매 시) [다른 판매자 보기\(2\)](#)

☒ 내일(월) 5/20 도착 보장 (15시간 10분 내 주문 시 / 서울경기 기준)

☐ 오늘(일) 도착 보장 ①

1 [장바구니 담기](#)
PC에서도 간편한 결제 쿠팡이머니 카드

모델명/품번: CM-SC01V USB

적립 최대 201원 쿠팡캐시 적립 · 쿠팡이머니 결제시

- 내/외장형 여부: 외장형
- 쿠팡상품번호: 5562811045 - 8829556851



조이트론

조이트론 유선 프로페셔널 핀마이크

★★★★★ 2,727개 상품평

22% 14,700원 ①

11,400원 로켓배송

무료배송 (로켓배송 상품 19,800원 이상 구매 시) [다른 판매자 보기\(3\)](#)

☒ 내일(월) 5/20 도착 보장 (15시간 9분 내 주문 시 / 서울경기 기준)

☐ 내일(월) 5/20 새벽 7시 전 도착 보장 (오후 3시 전 주문 시 / 서울경기 기준) ①

1 [장바구니 담기](#)
PC에서도 간편한 결제 쿠팡이머니 카드

모델명/품번
P5

적립 최대 570원 쿠팡캐시 적립 · 쿠팡이머니 결제시

- 유무선 여부: 유선
- 종류: 핀마이크
- 쿠팡상품번호: 260507696 - 814956987

새 상품 (3) / 반품 (1) 최저 9,120원 >

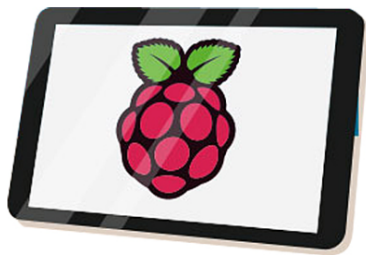
- 강의 내용 요약 시스템을 다른 학과 강의에 실제 적용하고 10명 이상의 평가단을 구성하여 평가를 진행하기 위해 추가 구매요청을 진행했습니다.

다른 학과의 강의실에 가서 직접 강의에 적용하여 평가를 진행하기 위해서는 휴대성이 있어야 된다고 생각했습니다.

사용자가 직접 들고 다니면서 강의 요약 시스템을 이용할 수 있도록 아래와 같은 제품들을 추가 구매하게 되었습니다.

[Raspberry Pi] 라즈베리파이 공식 7인치 터치스크린 (Raspberry Pi Touch Display)

Raspberry Pi 재단의 공식 Touchscreen Display 제품 / 해상도 800 x 480 / 라즈베리파이4,5 라즈베리파이3B+, 3B 호환



하트상품 1주 당일발송

상품번호 1273487

판매가(VAT 별도/포함) **85,000원** (93,500원)

평균준비기간 1~2일

제조사 Raspberry Pi

수량

- 1 +

1주의 라즈비안 OS Bullseye 정상 동작합니다. 최신OS - Bookworm에 이5에 호환됩니다.

총 상품금액 (VAT 별도)

[KKSBI] 라즈베리파이5 7인치 터치스크린용 메탈 케이스

한국공식대리점 / 라즈베리파이5 호환 7인치 터치스크린용 케이스 / 액티브 쿨러 장착 가능 / 완벽호환 / 깔끔한 마감 / 구성품: 케이스 x1, 디스플레이 케이블 x1 타 피스 / Case for Raspberry Pi 5 and the Official Raspberry Pi 7-inch Touchscreen



NEW 1주 당일발송

상품번호 15348490

판매가(VAT 별도/포함) **59,400원** (65,340원)

평균준비기간 1~2일

제조사 KKSBI

수량

- 1 +

1주의 보유재고 소진 시 약 2주 정도의 추가 납기가 소요될 수 있습니다.

총 상품금액 (VAT 별도)

4. 다음주(12주차) 일정

- 12주차 일정은 다음과 같습니다.

[5 월 20 일(월)]

- 19일에 배송 받은 사운드 카드와 마이크를 사용할 수 있게 세팅하고 작동을 확인.
- 선행적으로 진행중이던 STT부분 코딩의 최종 완료.

[5 월 21 일(화)]

- 요약알고리즘 데이터 분할 코드 작성.
- STT와 요약알고리즘 결합.

[5 월 22 일(수)]

- PPT 제작 및 주간보고서 작성.