

Project Schedule and Plan Presentation

프로젝트 일정 및 계획 프레젠테이션

배화여자대학교 SW보안융합학과
2024학년도 캡스톤디자인(2) A반 옴니버스

목차

Chapter 1

프로젝트 개요

Chapter 2

현행 구현 상태

Chapter 3

요구사항 변경

Chapter 4

향후 개발 계획

핵심 요약 및 결론

- 2024년 8월부터 구현 작업 진행 중
- 현재 (9.21. 기준) 프로젝트 완성도 약 57.1%
- 일부 요구사항 수정
- 하드웨어 모델, 형태소 분석기 · NLP 모델, 데이터베이스 사용 여부 등 변경됨
- NLP 모듈 구현 완료, 단위 테스팅 완료
- 입출력 컴포넌트 및 버스노선 검색 알고리즘 구현 중, 단위 테스팅 진행 중

음성 기반의 버스 노선 안내 시스템 개발

문제

- 노화 등의 사회 문화적 요소로 인한 스마트기기에 대한 접근과 활용에 제약이 있는 정보 취약계층
- 대중교통 이용 시 정보를 이용하려면 스마트기기에 능숙해야만 하는 정보 격차 문제 대두
- "지도 앱이라는 소프트웨어의 한계와 스마트폰이라는 기계의 한계를 극복할 수 있지 않을까?"

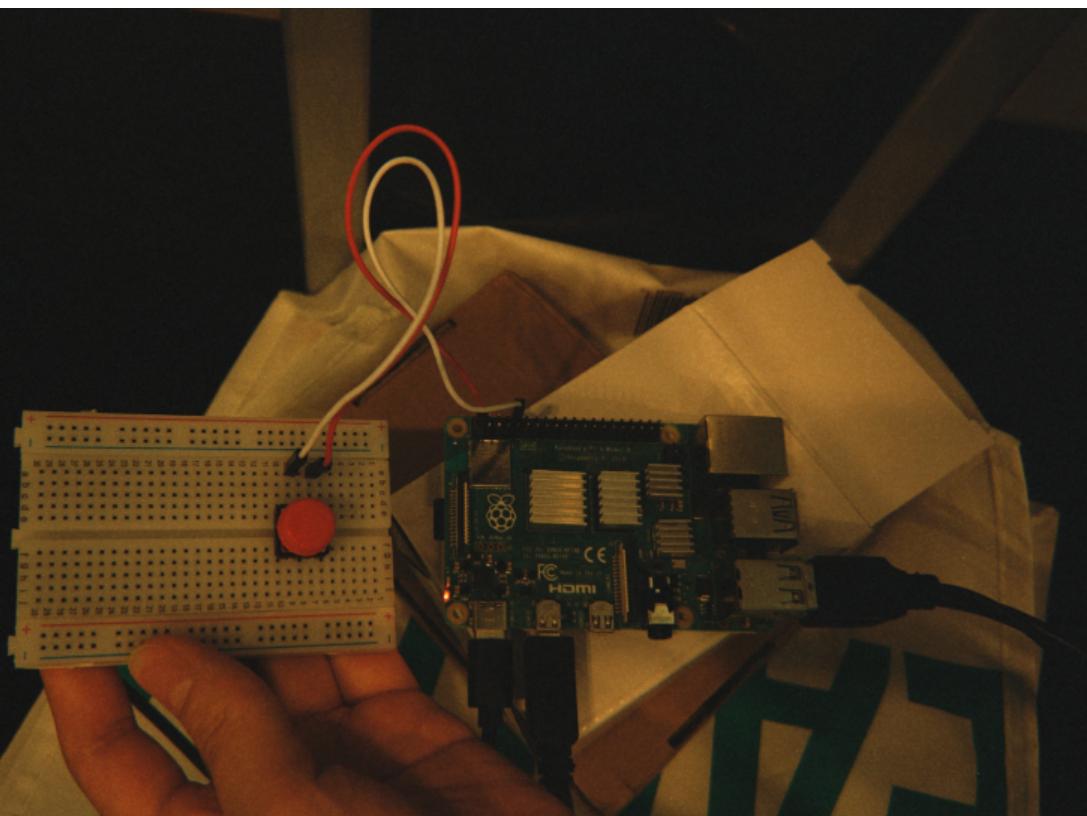


해결

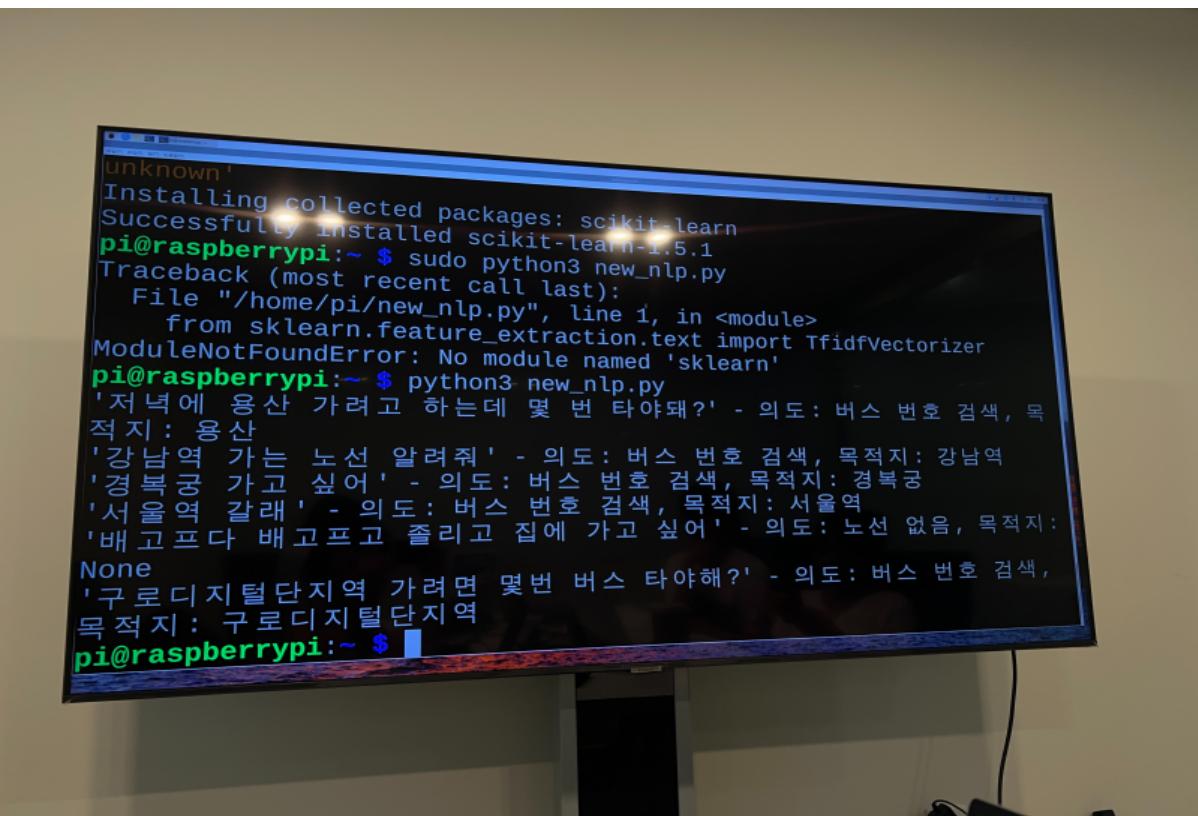
- 음성 기반의 버스 노선 안내 시스템(OMNIBUS, 옴니버스) 개발
- 사용자 경험(UX)을 극대하기 위한 단순한 디자인의 인터페이스 제작
- 음성 기반의 자연어 처리 기술 활용

현재 개발된 주요 기능

- I/O Componet의 음성 입력 및 출력 기능 구현 중
- Bus Route Searching의 버스 노선 검색 알고리즘 구현 중
- NLP 구현 완, 단위 테스팅 완



Raspberry Pi 개발 환경 세팅 완



NLP 모듈 단위 테스팅 수행 완

```

unknown'
Installing collected packages: scikit-learn
Successfully installed scikit-learn-0.5.1
Traceback (most recent call last):
  File "/home/pi/new_nlp.py", line 1, in <module>
    from sklearn.feature_extraction.text import TfidfVectorizer
ModuleNotFoundError: No module named 'sklearn'
pi@raspberrypi:~ $ python3 new_nlp.py
'저녁에 용산 가려고 하는데 몇 번 타야돼?' - 의도: 버스 번호 검색, 목적지: 용산
'강남역 가는 노선 알려줘' - 의도: 버스 번호 검색, 목적지: 강남역
'경복궁 가고 싶어' - 의도: 버스 번호 검색, 목적지: 경복궁
'서울역 갈래' - 의도: 버스 번호 검색, 목적지: 서울역
'배고프다 배고프고 즐리고 집에 가고 싶어' - 의도: 노선 없음, 목적지: None
'구로디지털단지역 가려면 몇번 버스 타야해?' - 의도: 버스 번호 검색,
목적지: 구로디지털단지역
pi@raspberrypi:~ $ [REDACTED]

```

Raspberry Pi 환경에서 단위 테스팅 수행 중

이곳에는 원인과 결과를 요약해 입력하세요.

원인

- 먼저 현상이나 **문제점을 명확하게 설명합니다.**
- 어떤 **현상이 발생했는지를 상세히 기술합니다.**
- 현상이 발생한 원인에 대한 여러 가지 가설을 제시해 보세요.
- 가능한 원인을 명시하고 각각에 대한 근거 또한 제시해요.

결과

- 해당 원인을 해결하거나 예방하기 위한 **향후 조치나 대응 방안을 제안할 수 있습니다.**
- 결과의 관리나 개선을 위한 제안을 포함해도 좋아요.

GitHub Repository

- URL: <https://github.com/river20s/OMNIBUS>

The screenshot shows the GitHub repository page for 'OMNIBUS'. The repository was created by 'CheulSu' on March 1, 2024. It has 1 branch and 0 tags. The repository contains 6 commits. The code tab is selected. The repository structure includes 'configs', 'data', 'scripts', 'src', 'tests', '.gitignore', 'README.md', and 'requirements.txt'. The README file is titled '온니버스 OMNIBUS (Raspberry Pi)' and describes it as a project for a bus information system. Contributors listed are river20s, Subin-Lim, hana021800, and CheulSu.



QR code

1. 하드웨어 모델 업그레이드

기존

>>>

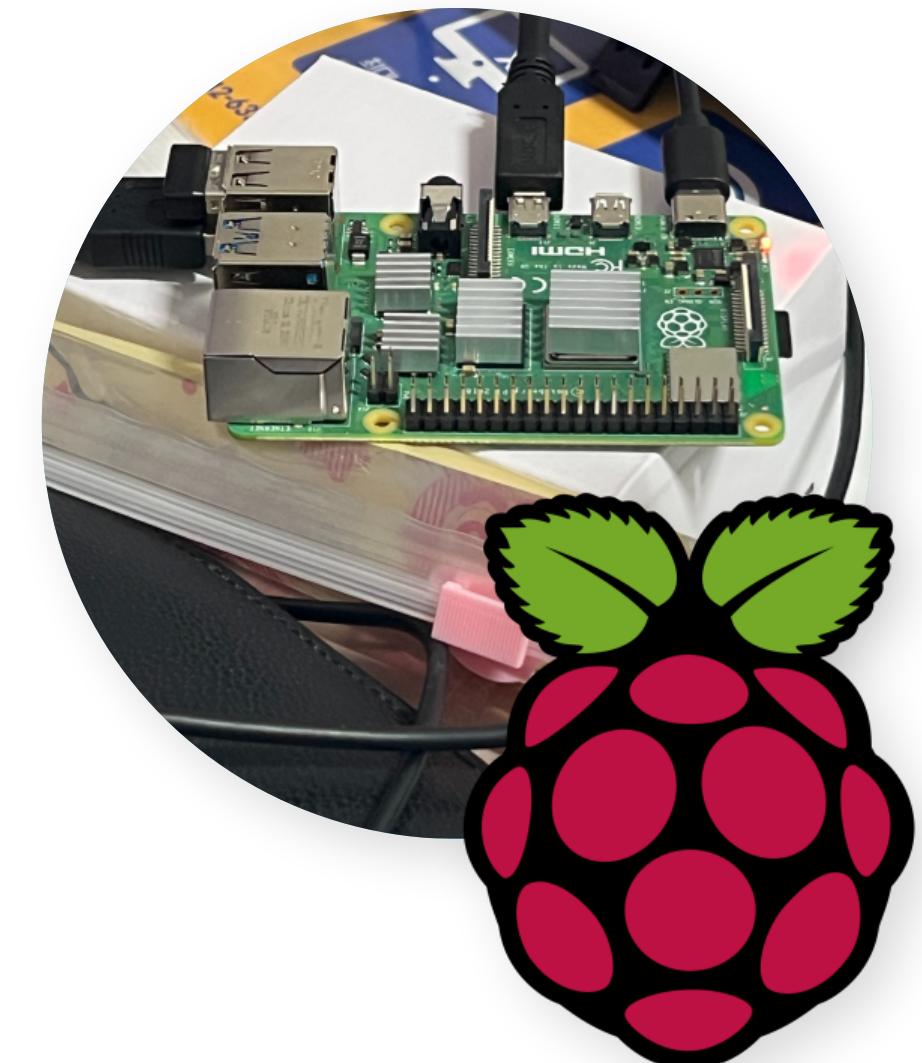
변경 후

Raspberry Pi 4B 2GB

기존 NLP 코드 실행 시
메모리 부족 현상이 발생함
스왑 파일 크기 조정, 가벼운
모델 사용, 원격 실행 등 시도
한 해결 방안 모두 실패

Raspberry Pi 4B 4GB

보다 큰 RAM 용량을 가진
모델로 업그레이드, 경량화
버전 모델 사용, 원격 실행
세 방법을 사용하여 단위
테스트 수행 통과함



2. 경량화 된 형태소 분석기 사용

기존

Komoran

차지하는 메모리 크기를
확보하기 어려워 메모리 부족
현상 발생, NLP 코드 자체가
Raspberry Pi 안에서
실행되지 않음

>>>

변경 후

Okt Open Korean Text

보다 가벼운 형태소 분석기인
Okt로 변경, 주어진 데이터에
서 간단한 한국어 처리를 통해
명사만 추출하는 방식으로
텍스트 전처리 방식 변경

```

from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.svm import SVC
from konlpy.tag import Okt

class SimpleIntentDetector:
    def __init__(self):
        # 문서를 파일로 쓰도록 설정
        self.okt = Okt()
        # 텍스트 파일을 TF-IDF SVM 모델
        self.vectorizer = TfidfVectorizer()
        self.model = SVC(probability=True)

    # 텍스트 데이터셋
    self.train_texts = [
        "제주 가야도 빙판 천 번의 일",
        "강릉에 가는 노년 청와대",
        "고령군 가고 싶어",
        "서울에 살다",
        "제주 가고 싶어",
        "제주 가고 싶어",
        "제주 가고 싶어"
    ]
    self.train_labels = [
        "제주 관광 목적",
        "제주 관광 목적"
    ]
    # 텍스트 학습과 평가
    self.train()

    def preprocess_text(self, text):
        # 형태소 분석 후, 텍스트 속 토큰이 리스트로 반환
        tokens = self.okt.nouns(text)
        return ' '.join(tokens)

    def train(self):
        # 텍스트 학습
        processed_texts = [self.preprocess_text(text) for text in self.train_texts]
        X_train = self.vectorizer.fit_transform(processed_texts)

        # SVM 학습
        self.model.fit(X_train, self.train_labels)

    def extract_destination(self, text):
        # 목적지 분석
        tokens = self.okt.pos(text)

        # 목적지 주체 (고유한) Noun 또는 *의*(의 모형문 형태를 제외하고) 간주
        destination = None
        for token, pos in tokens:
            if pos == 'Noun' and '의' in token: # "의"의 포함은 명사를 제외하고 고려
                destination = token
            else:
                if pos == 'Verb':
                    intent = self.recognize_intent(text)
                    if intent == '제주 관광 목적':
                        if destination == None:
                            destination = token
                    else:
                        destination = None
                else:
                    if destination == None:
                        destination = token
                    else:
                        if pos == 'Noun' and len(token) > 1 and token not in ['제고도', '제자', '제고도', '제자']:
                            destination = token
        return destination

    def recognize_intent_and_destination(self, text):
        # 목적지 추출
        destination = self.extract_destination(text)

        # 고유한 또는 "의" 포함된 경우 '버스' 번역 '으로' 붙임
        if destination:
            intent = '제주 관광 목적'
        else:
            intent = self.recognize_intent(text)
            if intent == '제주 관광 목적': # 잘못된 예측일 경우 '노선 찾기'으로
                intent = '노선 찾기'

        return intent, destination

    def recognize_intent(self, text):
        # 목적지 추출
        processed_text = self.preprocess_text(text)

        # TF-IDF 적용
        X_input = self.vectorizer.transform([processed_text])

        # 모델 적용
        intent = self.model.predict(X_input)[0]
        return intent

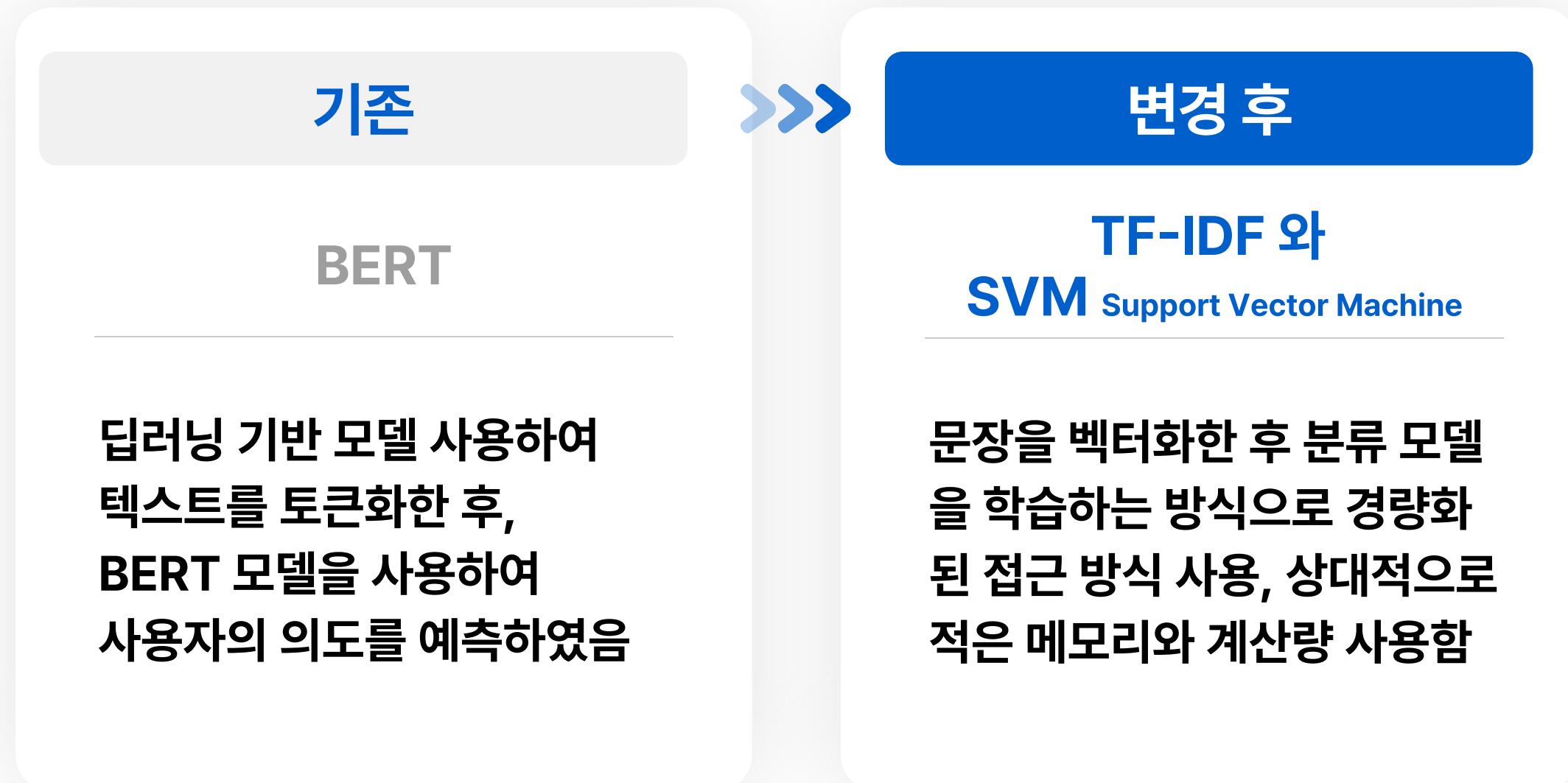
    # 텍스트 검색
    intent_detector = SimpleIntentDetector()
    texts = [
        "제주에 봄은 가려고 했는데 몇 번의 일",
        "강릉에 가는 노년 청와대",
        "고령군 가고 싶어",
        "제주에 살다",
        "제주 가고 싶어",
        "제주 가고 싶어",
        "제주 가고 싶어"
    ]

    for text in texts:
        intent, destination = intent_detector.recognize_intent_and_destination(text)
        print(f'{text}: {intent}, {destination}')

```



3. NLP 모델 변경

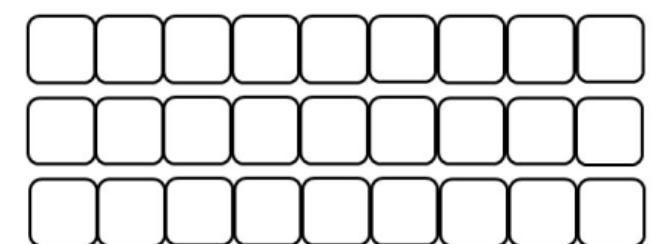


TF-IDF

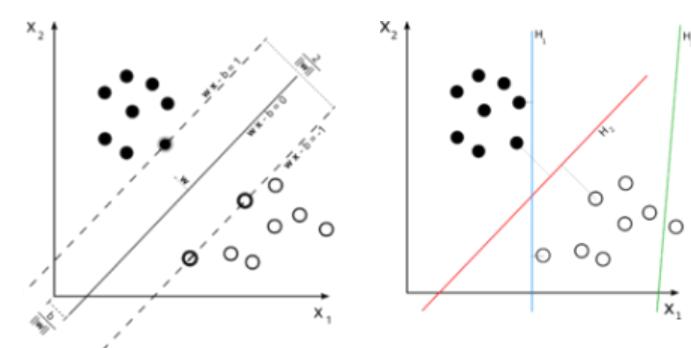


$$tf(t, d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}$$

$$idf(t, D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$



이미지 출처: Youtube | DataMListic



이미지 출처: 위키피디아 | 서포트 벡터 머신

4. 의도 인식과 목적지 추출을 위한 NLP 아키텍처 개선

기존

>>>

변경 후

변경 후

사전 학습된 BERT 모델 사용,
별도의 학습 과정 없이 미리
학습된 모델을 사용해 바로
예측하여 작업 수행함

TfidVectorizer와 SVM을
사용해 사용자 정의 학습 데이
터셋으로 모델 훈련, 텍스트는
형태소 분석을 통해 명사만 추
출하여 학습 후 SVM 분류
모델 학습

목적지 추출 과정에 불필요한
단어 필터링 추가, 예측된
의도는 목적지 추출 여부에
따라 '버스 번호 검색' 또는
'노선 없음'으로 조정됨

5. 데이터베이스 사용 여부

기존

>>>

변경 후

사용자가 입력한 목적지에
대해 SQL 쿼리 실행하여
해당 목적지와 일치하는
버스 노선 데이터를 검색
하는 방식

공공 데이터 API의 버스 노선
검색 기능, Google Maps
API의 장소 검색 기능을
활용하여 정보를 조회하는
방식으로 전환, 더 다양한
데이터 소스 접근 가능하며
실시간 정보 제공 가능하도록
기능 확장함



Google Places API

DATA . GO . KR
공공데이터포털

* 2024년 9월 21일 기준

		8			9			10				11	담당자		
개발 업무		10	17	24	31	7	14	21	28	5	12	19	26	-	
Application	I/O Component	STT, TTS API 연동													남현서
	I/O 인터페이스														
	화면 표시 알고리즘														
	오류 처리 및 개선														
NLP	NLP	모델 및 토크나이저 초기화													임수빈
		텍스트 전처리	토큰화												
		알고리즘	품사태깅												
		의도 인식 알고리즘	개체 인식												
	Bus Route Searching	목적지 추출 알고리즘													
		오류 처리 및 개선													
		Google Maps, 공공데이터 API 설정													
Hardware	Raspberry Pi	위치 좌표 변환 알고리즘													임하나
		검색 알고리즘													
		오류 처리 및 개선													
		통합 인터페이스													
	외관 디자인	개발 환경 세팅													양수연
Application	Application Testing	단위 테스트													강은빈
		통합 테스트													
		시스템 테스트													
		외관 디자인													

