



그놈목소리

한양대학교 이지훈 권현서 길미주

진행 순서

상황분석

01

아이디어 도출 배경

아이디어 소개

02

프로토타입

데이터 확보

03

데이터 크롤링 및 전처리 사용된
모든 코드는 Github에 기록됨

모델링

04

ETRI NER 활용 보이스 피싱 탐지
모델링 사용된 모든 코드는 Github에 기록됨

확장

05

아이디어 강점 및 발전 방향

1

상황 분석



언택트 시대에 더 늘어난 보이스피싱



코로나 휩쓴 작년 보이스피싱 취약계층 노렸다...6년만에 최대

금감원 접수 보이스피싱 피해 신고만 5만2165건...전년비 60% 급증

(서울=뉴스1) 서상혁 기자 | 2021-05-30 12:00 송고



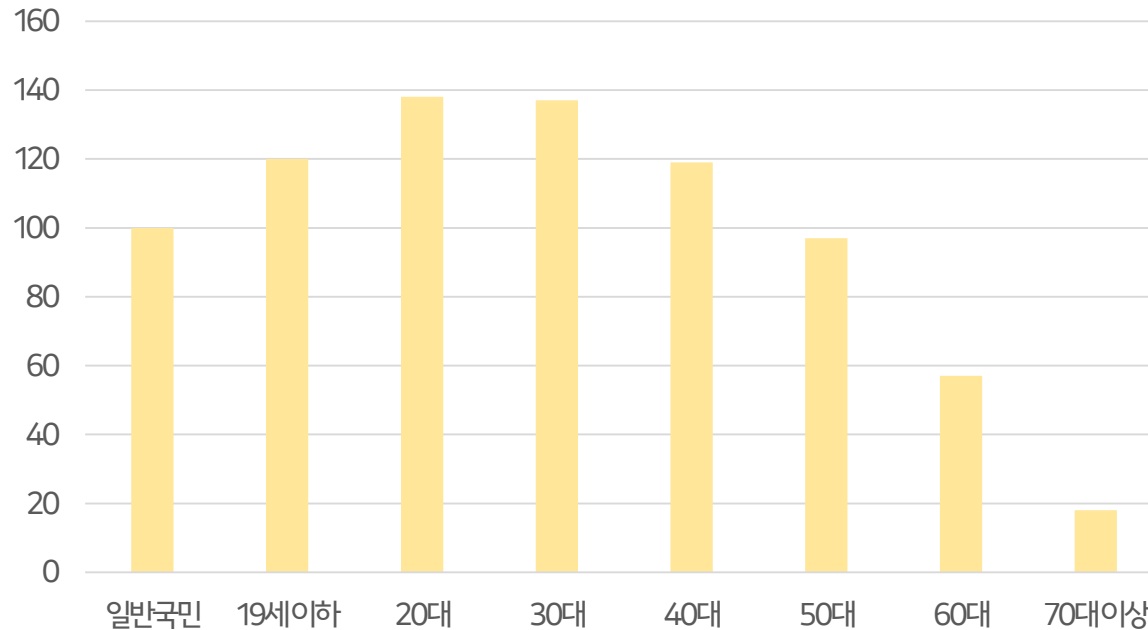
지난해 보이스피싱 피해액은 약 **2300억**

보이스피싱은 전년대비 신고 건수가 **60.7% 증가**

코로나19 장기화에 경제적 어려움을 겪는 서민을 대상으로 한 불법 금융행위가 급증
AI 기술 혁신을 바탕으로 고도화되는
금융 사이버 위협을 선제적, 전략적으로 막을 필요가 대두되고 있음

언택트 시대에 더 늘어난 보이스피싱

<연령별 디지털 정보화 역량 수준>



일반 국민의 디지털 정보화 수준을 100으로 할 때,
고령층의 디지털 정보화 종합 수준은 64.3%에 불과함

더욱 가속화된 디지털 금융환경 변화에 고령층 등 취약계층이
소외되지 않도록 다양한 지원방안 마련의 필요성 증가

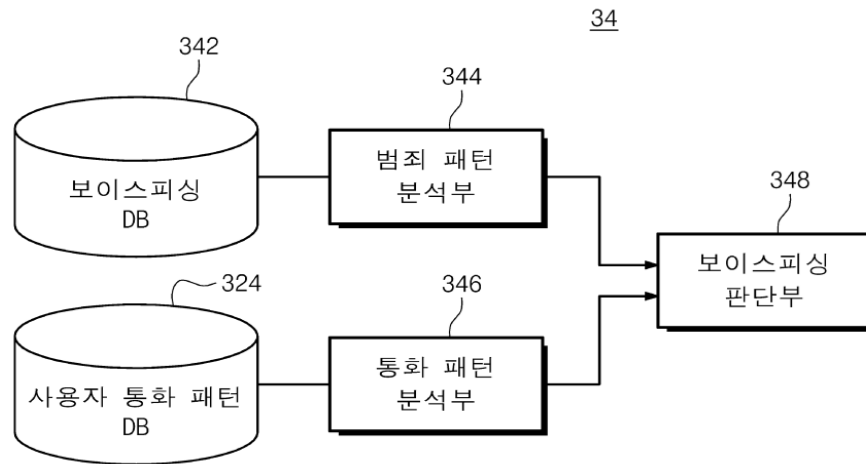
또한, 코로나19로 촉발된 국제적인 경제위기의 후유증으로
경제 불평등 심화 및 취약계층의 금융 소외 현상 계속해서 증가

관련 연구 (특허)

보이스피싱탐지방법,보이스피싱탐지장치및기록매체

특허출원번호: 10-2018-0172005, 출원일 2018년 12월 28일, 공개일 2020년 7월 8일

대표도 - 도3

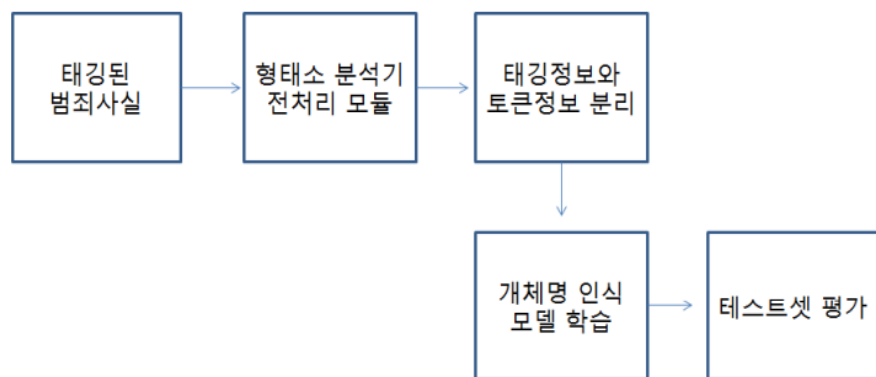


특허내용: 사용자와 상대방간의 통화내용을 양방향 모니터링하여
보이스피싱을 탐지하는 보이스피싱 탐지장치

- 1) 사용자별로 통화습관을 학습하여 정상통화패턴을 저장
- 2) 상대방의 전화번호가 블랙리스트나 다른 리스트인지 판단
- 3) 판단한 리스트를 기반으로 통화내용으로부터 상대방의 범죄패턴을 분석하여 보이스피싱 탐지

관련 연구 (논문)

인공지능 기반 개체명 인식 모델의 보이스피싱 여죄 분석 활용에 관한 연구
김희두 外 3인, 경찰대학 경찰학연구편집위원회, 경찰학연구, 경찰학연구 제20권 제4호



〈그림 1〉 개체명 인식 실험 개요도

논문 내용: 개체명 인식 모델(NER)을 활용해
과거 보이스피싱 DB에서 사건의 여죄를 분석함

- 1) 보이스피싱 수법으로 분류된 범죄사실에서 사칭기관, 사칭이름, 범행이용 계좌의 은행명 등을 자동으로 추출하는 NER 연구 진행
- 2) 개체명 인식에서 특히 성능이 높다고 알려진 Bidirectional LSTM-CRF 모델을 사용해 보이스피싱 확인에 사용

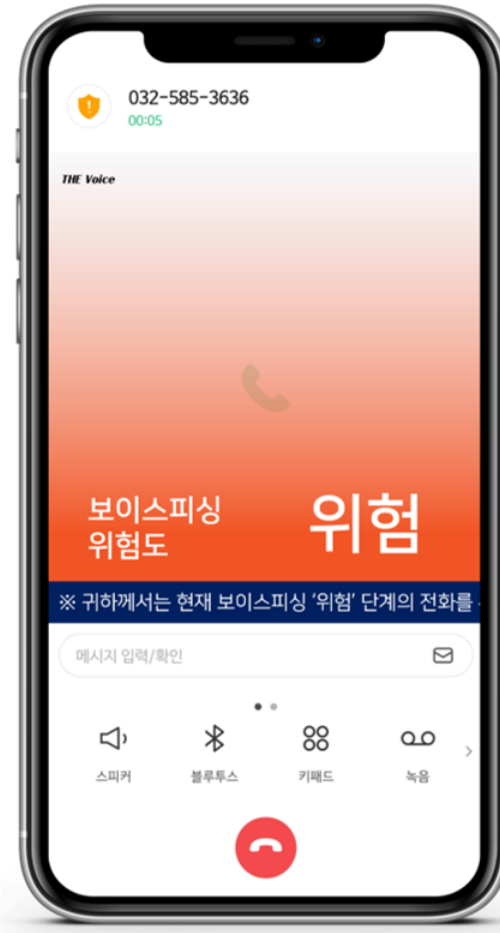
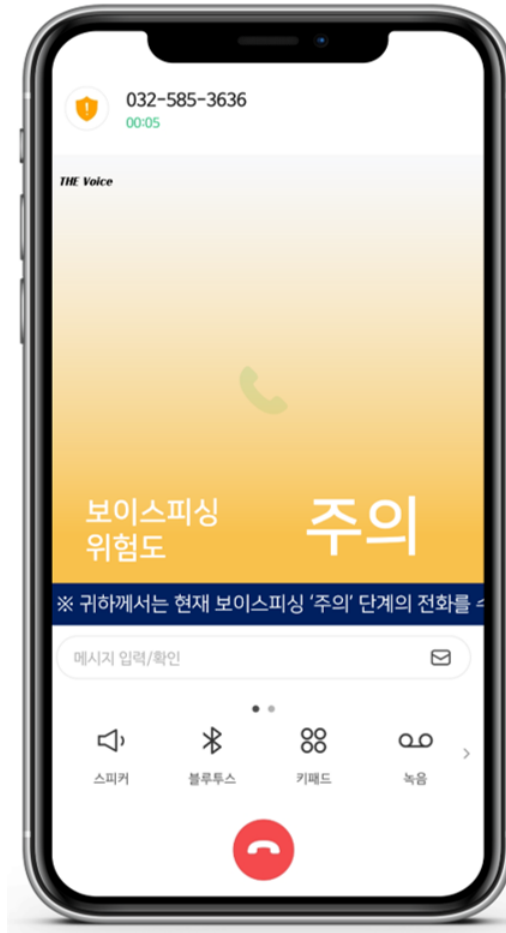
2

아이디어 소개



개체명 인식 기반 보이스피싱 탐지 모델

실시간 전화상황에서 대화내용 텍스트화(ETRI STT) 및
내용 텍스트의 개체명 인식(ETRI NER)을 통해 보이스피싱 위험도를 실시간으로 게재



기존 특허 / 논문 대비 차별점



<특허대비강점>

보이스피싱의 실시간 탐지에 NER 태깅 모델과 문장 N-gram 언어 모델을 활용하여
통화 패턴에 따른 보이스피싱 여부 파악이 가능하며, 사용자에게 실시간으로 보이스피싱 탐지 안내

<논문 대비 강점>

대용량의 말뭉치를 사전 학습한 언어 모델을 이용하여 소규모 데이터에서도 적합하게 활용했으며
전체적인 모델링 과정에서 NER, 문장 N-gram, Sentence Transformer 등을 사용함으로써 완성도를 높임

특장점1) 대화 흐름을 학습

문장 N-gram으로 변화하는 대화 흐름을 이해하고 탐지함으로써
기존의 rule-based 판단 기법보다 전화 특성을 반영한 예측 가능

특장점2) 범죄 유형 변화에 능동적 적응

NER로 변화하는 범죄 유형에 능동적으로 적응함으로써
빠르게 증가하는 신종 보이스피싱 수법을 예측하고 그 피해를 방지

3 데이터 확보

3-1. 보이스피싱 데이터



보이스피싱 데이터 확보

금융감독원 보이스피싱 지킴이사이트의 보이스피싱 체험관에서 제공하는 '대출사기형', '수사기관사칭형' 보이스피싱 텍스트 데이터 크롤링 진행

제목	등록일
KB저축은행입니다. 핸드폰 어떤거 사용하세요?	2020-12-10
우리은행 직원인데요 영업점에선 절 모를수도있어요	2020-12-10
KB저축은행인데요 카카오톡친구추천부탁드립니다	2020-12-10
영업점에선 절 모를수도있어요(...???)	2020-12-10
OO캐피탈입니다. 심사결과 확인되어서 연락드렸습니다.	2020-12-07
OO저축은행입니다.	2020-12-07
동 상품은 연체를 하시거나 고의적으로 상환을 회피하실 경우 고객님의 미래 소득에 대한 추심권한을 부여받고 진행되는 대출상품입니다. 대출 서류 접수를 위해 사업자등록증과 신분증 사본을 보내주세요	2018-06-14
고객님 상환하셨다는 서류를 고객센터로 보내야 하는데, 수요일이 공휴일이라 처리가 밀려서 아직 상환내역이 확인안되시는 걸거예요	2018-06-14
(통장매매) 사장님께서 안쓰시는 계좌 있으시면 월 400~450만원까지 선 지급해드리고 임대하고 있어 연락 드렸습니다	2018-06-14
현재 이용하시는 고금리 대출 상품들 저금리 전환 가능하세요. 현재 이용하시는 고금리 대출이 얼마정도 되시나요?	2018-06-14

<대출사기형>

제목	등록일
금융범죄수사중입니다. 사건내용을 유출할 경우 처벌받을수있습니다	2020-12-10
본인과 연루된 불법 명의도용사건이 검찰에 접수되어 연락드렸습니다	2020-12-10
다음주에 재판이 열립니다. 재판증거자료를 위해 통화를 녹취하겠습니다.	2020-12-10
강남서 사이버수사과입니다	2020-12-07
이 사건은 전자금융거래법 위반 금융사기 사건으로, 다수의 피해자가 발생한 고소고발 사건입니다	2018-06-14
이 전화는 녹취전화라 수신이 안되므로 혹시 향후 궁금한 점 생기시면 서울중앙지검에서 'OOO 수사관'을 찾아주세요	2018-06-14
(동일 사기범 다수 제보 접수 건 편집본) 저는 서울중앙지검 강력수사팀 000 수사관입니다. 명의가 유출되어 피해를 보신 부분에 대해서는 보상을 받으실 수 있습니다.	2018-06-12
(동일 사기범 다수 제보 건 편집본) 녹취라고 해서 별다른 건 없고, 아까 제가 드렸던 질문을 그대로 녹취에 담을 겁니다. 수사 협조 부탁드립니다	2018-06-12
(동일 사기범 다수 제보 건 편집본) 귀하의 이름을 판매자로 등록한 후 고가의 항공권, 상품권을 판다고 한 후, 돈만 가로채고 도주한 사건입니다	2018-06-12
(동일 사기범 다수 제보 건 편집본) 중고나라 사이트를 통해 고가의 물건을 등록해놓고 물건은 양도하지 않고 자금만 편취한 사건인데, 금융범죄사기로 분류되어 저희 첨단범죄수사부에서 사건 진행을 맡고 있습니다	2018-06-12

<수사기관사칭형>

보이스피싱 데이터 확보

각 게시글에는 실제 보이스피싱 음성 파일과,
음성파일이 텍스트로 변환된 형태가 선택적으로 실려 있었음

▶ 0:00 / 4:57 ———— 🔊 ⋮

사기범 : 여보세요
피해자 : 네
사기범 : 네 안녕하세요, 혹시 000 핸드폰 맞으십니까?
피해자 : 어디세요?
사기범 : 예 수고하십니다, 서울 중앙 지검 검찰청입니다.
피해자 : 네
사기범 : 예 본인 앞으로 몇 가지 확인 사항 때문에 전화 드렸는데요.
피해자 : 네

모든 게시글에서 텍스트에 해당하는 부분을
Selenium Webdriver로 크롤링 및 텍스트 전처리

보이스피싱 데이터 전처리

대출사기형 총 19페이지, 수사기관사칭형 총 23페이지를 크롤링해
['주소', '내용']의 컬럼을 가진 데이터프레임을 반환함

```
voicephishing_대출사기형 = get_대출사기형_info()  
#voicephishing_대출사기형 = pd.read_csv('voicephishing_대출사기형.csv', index_col=0)
```

```
voicephishing_대출사기형.head(10)
```

	주소	내용
0	https://phishing-keeper.fss.or.kr/fss/vstop/bb...	[]
1	https://phishing-keeper.fss.or.kr/fss/vstop/bb...	[]
2	https://phishing-keeper.fss.or.kr/fss/vstop/bb...	[]
3	https://phishing-keeper.fss.or.kr/fss/vstop/bb...	[]
4	https://phishing-keeper.fss.or.kr/fss/vstop/bb...	[]
5	https://phishing-keeper.fss.or.kr/fss/vstop/bb...	[]
6	https://phishing-keeper.fss.or.kr/fss/vstop/bb...	['피해자 : 여보세요', '사기범 : 네 (빠-) (빠-) 님', '피해자 : 예...
7	https://phishing-keeper.fss.or.kr/fss/vstop/bb...	['피해자:여보세요', '사기범:안녕하세요 고객님 김종현 대리입니다', '피해자:아...
8	https://phishing-keeper.fss.or.kr/fss/vstop/bb...	['사기범:네 여보세요?', '피해자:여보세요', '사기범:네 ㅇㅇ 맞으신가요?',...
9	https://phishing-keeper.fss.or.kr/fss/vstop/bb...	['피해자:네', '사기범:네 여보세요', '피해자:뭐 어디세요?', '사기범:아 ...

데이터를 제대로 활용하고자
전처리가 요구되었음

<크롤링으로 얻은 '대출사기형' 텍스트 데이터>

보이스피싱 데이터 전처리

전처리규칙1) 텍스트 데이터를 포함하지 않는 row는 드랍

```
# 텍스트 있는 경우만 남김
voicephishing_대출사기형_text = list(voicephishing_대출사기형['내용'])
voicephishing_대출사기형_text = [i for i in voicephishing_대출사기형_text if i != '[]']
len_대출사기형_text = len(voicephishing_대출사기형_text)
print(f'대출사기형 보이스피싱 텍스트 건수: {len_대출사기형_text}')
```

전처리규칙2) 피해자, 사기범을 가리키는 명칭 텍스트 제거

```
# 피해자/사기범 명칭 제거
for i in range(len_대출사기형_text):
    cur_text = voicephishing_대출사기형_text[i]

    for j in range(len(cur_text)):
        # 피해자 명칭 제거
        if ('피해자:' in cur_text[j]) or ('피해자 : ' in cur_text[j]):
            cur_text[j] = re.sub(r'피해자:', '', cur_text[j])
            cur_text[j] = re.sub(r'피해자 : ', '', cur_text[j])
            voicephishing_대출사기형_text[i][j] = cur_text[j]
            continue

        # 사기범 명칭 제거
        if ('사기범:' in cur_text[j]) or ('사기범 : ' in cur_text[j]):
            cur_text[j] = re.sub(r'사기범:', '', cur_text[j])
            cur_text[j] = re.sub(r'사기범 : ', '', cur_text[j])
            voicephishing_대출사기형_text[i][j] = cur_text[j]
```

전처리 전 예시: '피해자:여보세요', '사기범:안녕하세요 고객님 김종현 대리입니다'

전처리 후 예시: '여보세요', '안녕하세요 고객님 김종현 대리입니다'

보이스피싱 데이터 전처리

전처리규칙 3) 기존 텍스트를 문장단위로 잘라냄

```
# 문장 분리
for i in range(len_대출사기형_text):
    cur_text = voicephishing_대출사기형_text[i]
    splited_text = []

    for j in range(len(cur_text)):
        # 문장 분리 후 길이 저장
        splited = split_sentences(cur_text[j])
        len_splited = len(splited)

        # 길이가 1이면 바로 추가함
        if len_splited == 1:
            splited_text.append(cur_text[j])
        # 길이가 1이 아니면 나눠서 추가함
        else:
            for k in range(len_splited):
                splited_text.append(splited[k])

    # 분리한 문장으로 대체
    voicephishing_대출사기형_text[i] = splited_text
```

전처리 전 예시: '네 000씨 지금 제가 본인 확인 한번 할게요. 생년월일 앞에 여섯 자리만 말씀 부탁드립니다.'

전처리 후 예시: '네 000씨 지금 제가 본인 확인 한번 할게요.', '생년월일 앞에 여섯 자리만 말씀 부탁드립니다.'

전처리규칙 4) 텍스트에 욕설이 등장하면 보이스피싱 상황이 종료된 것으로 간주하여, 욕설이 등장한 부분부터 해당 사건 텍스트의 맨 끝까지 내용을 모두 제거

```
# 욕설 제거
for i in range(len_대출사기형_text):
    cur_text = voicephishing_대출사기형_text[i]

    for j in range(len(cur_text)):
        # 욕설 이후 제거
        if 'xx' in cur_text[j]:
            del voicephishing_대출사기형_text[i][j:]
            break
```

전처리 예시: 대화 텍스트 중 'xx'가 등장하면 해당 문장 포함 그 이후 모두 제거

보이스피싱 데이터 최종

보이스피싱 데이터 크롤링 및 전처리 결과

```
# 문장당 카운트 횟수
countsent_대출사기형 = 0
for i in range(len_대출사기형_text):
    countsent_대출사기형 += len(voicephishing_대출사기형_text[i])
print(f'대출사기형 사기범 문장 개수: {countsent_대출사기형}')

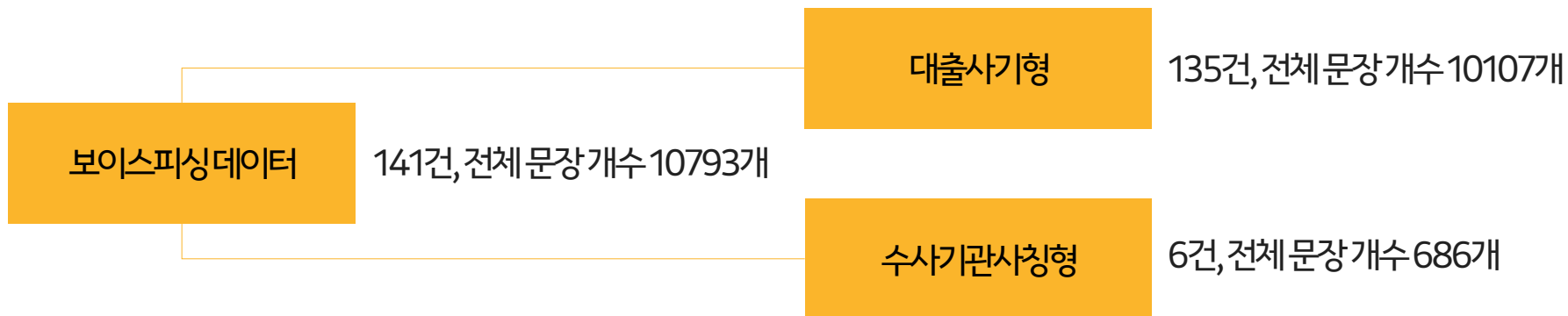
countsent_수사기관사칭형 = 0
for i in range(len_수사기관사칭형_text):
    countsent_수사기관사칭형 += len(voicephishing_수사기관사칭형_text[i])
print(f'수사기관사칭형 사기범 문장 개수: {countsent_수사기관사칭형}')
```

대출사기형 사기범 문장 개수: 10107
수사기관사칭형 사기범 문장 개수: 686

```
# 파일 통합
voicephishing_전체_text = voicephishing_대출사기형_text + voicephishing_수사기관사칭형_text
len_전체_text = len(voicephishing_전체_text)
print(f'보이스피싱 전체 사건 개수: {len_전체_text}')

countsent_전체 = 0
for i in range(len_전체_text):
    countsent_전체 += len(voicephishing_전체_text[i])
print(f'전체 문장 개수: {countsent_전체}')
```

보이스피싱 전체 사건 개수: 141
전체 문장 개수: 10793



3 데이터 확보

3-2. 콜센터 데이터





콜센터 데이터 확보

AI Hub에서 제공하는 '상담 음성'에서
'금융 도메인'에 해당하는 텍스트 데이터를 이용함

상담 음성

소개

다운로드

음성/자연어
상담 음성

텍스트 오디오

2020

데이터셋명	상담 음성			
데이터 분야	음성/자연어	데이터 유형		텍스트, 오디오
구축기관	티맥스소프트	데이터 관련 문의처	담당자명	임성민(티맥스소프트)
가공기관	아이스크림에듀		전화번호	031-8018-1821
검수기관	나무기술		이메일	seongmin_lim@tmax.co.kr
구축 데이터량	432만	구축년도		2020년
버전	1.1	최종수정일자		2021.09.09
소개	웹 기반 등 다양한 방식으로 상담센터에 연락하여 상담하는 내용을 녹음한 음성 데이터			
주요 키워드	AI 상담사, AI 콜센터, 한국어 음성언어처리 기술, 교육 도메인, 금융 도메인, 통신판매 도메인, 음성 품질, 명료도, 대화 주제, 저작권			
저작권 및 이용정책	본 데이터는 과학기술정보통신부가 주관하고 한국지능정보사회진흥원이 지원하는 '인공지능 학습용 데이터 구축사업'으로 구축된 데이터입니다 [데이터 이용정책 상세보기]			

콜센터 데이터 확보

크롤링을 통해 얻은 보이스피싱 데이터가 총 141건, 문장 10725개인 것에 비해
AI Hub의 상담 음성 중 금융 도메인 데이터는 전체가 16771건, 문장 456165개라 다운 샘플링이 불가피했음

상담데이터의 문장개수가 보이스피싱 데이터 문장개수의
오차범위 +/- 1%에 해당하도록 다운 샘플링 진행함으로써
서로 다른 두 데이터의 문장수를 거의 동일한 수준으로 맞춰줌

보이스피싱	10793
상담	456165

<다운샘플링전문장개수>

```
# 보이스피싱 건수와 맞게 다운 샘플링
while(not success):
    before = dataframe
    randnum = random.randrange(1, 1501)

    # 건수를 200개로 다운 샘플링
    down_sampled = before.sample(n=200, random_state=randnum).reset_index(drop=True)
    down_sampled_count = list(down_sampled['문장 개수'])
    print(f' 다운 샘플링 된 전체 문장 개수: {sum(down_sampled_count)}')

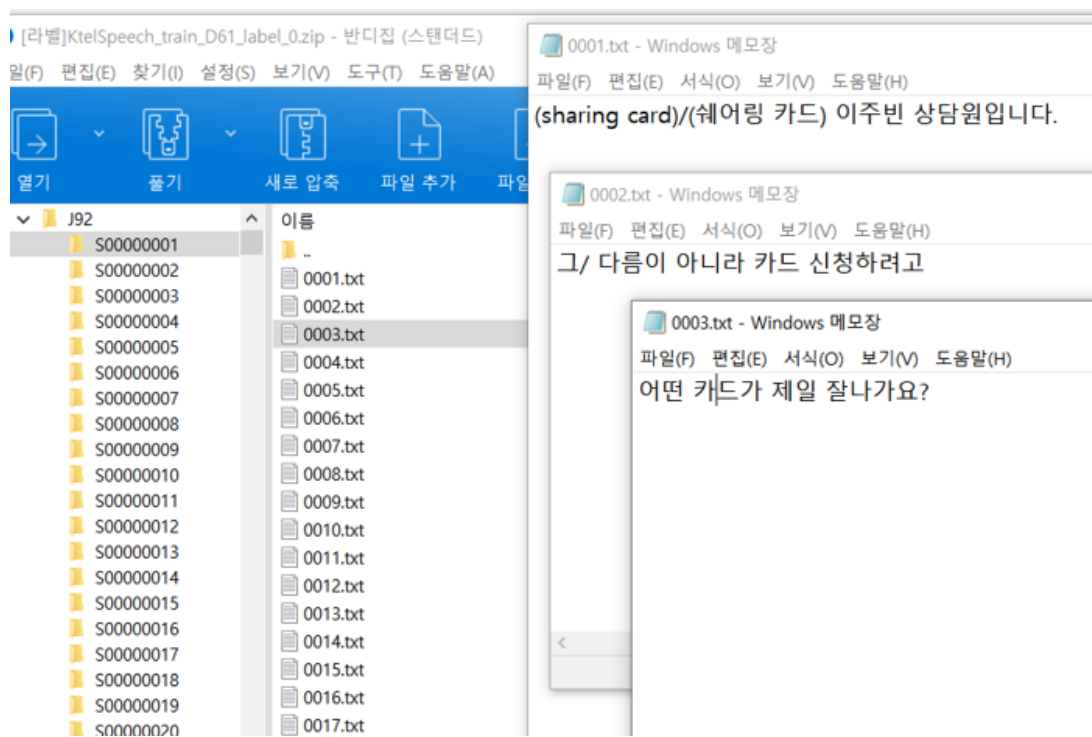
    # 보이스피싱 건수의 오차범위 +/- 1%에 들면 성공
    if (10793*0.99 <= sum(down_sampled_count)) and (sum(down_sampled_count) <= 10793*1.01):
        print(' 다운 샘플링 성공')
        success = True
```

보이스피싱	10793
상담	10725

<다운샘플링후문장개수>

콜센터 데이터 전처리

AI Hub에서 제공하는 데이터에는 건당 문장이
개별 txt파일로 저장되어 있어, 이를 한 파일로 통합할 필요가 있었음



<AI Hub상담 음성데이터 중 금융 도메인 텍스트 데이터>

데이터를 제대로 활용하고자
문장 통합 및 전처리가 요구되었음

콜센터 데이터 전처리

전처리규칙 1) 기존 텍스트를 문장단위로 잘라냄

```
# 문장 분리
for i in tqdm(range(len(상담_text))):
    cur_text = counseling_전체_text[i]
    splited_text = []

    # 문장 분리 후 길이 저장
    splited = split_sentences(cur_text)
    len_splited = len(splited)

    # 문장을 나뉘서 추가함
    for j in range(len_splited):
        splited_text.append(splited[j])

    # 분리한 문장으로 대체
    counseling_전체_text[i] = splited_text
```

전처리규칙 2) 데이터 원문에 포함되어 있었던 특이한 텍스트 형태를 올바르게 정제

```
for j in range(len(cur_text)):
    # 개행문자 제거
    cur_sentence = cur_text[j]
    cur_sentence = re.sub(r'\n| ', '', cur_sentence)

    # + 표현 제거
    cnt = 0
    matches0 = re.finditer(r'#+', cur_sentence)
    for match in matches0:
        cnt += 1

    while cnt > 0:
        matches1 = re.finditer(r'#+', cur_sentence)
        for match in matches1:
            plus_index = match.span()[0]
            break
        plus_index_saved = plus_index

        blank_index = -1
        while plus_index >= 0:
            if cur_sentence[plus_index] == ' ':
                blank_index = plus_index
                break
            else:
                plus_index -= 1

        if blank_index == -1: blank_index = 0
        dropword = cur_sentence[blank_index:plus_index_saved+1]
        cur_sentence = cur_sentence.replace(dropword, '')
        cnt -= 1

    # ()/() 표현 제거
    cnt = 0
    matches0 = re.finditer(r'#[^)]+#[^)]+#[^)]+', cur_sentence)
    for match in matches0:
        cnt += 1

    while cnt > 0:
        matches1 = re.finditer(r'#[^)]+#[^)]+#[^)]+', cur_sentence)
        for match in matches1:
            start_idx = match.span()[0]
            break

        matches2 = re.finditer(r'#[^)]+#[^)]+', cur_sentence)
        for match in matches2:
            center_idx = (match.span()[0] + match.span()[1]) // 2
            break

        dropword = cur_sentence[start_idx:center_idx+1]
        cur_sentence = cur_sentence.replace(dropword, '')
        cnt -= 1

    cur_sentence = re.sub(r'/', '', cur_sentence)
    cur_sentence = re.sub(r'#[^)]+', '', cur_sentence)
    cur_sentence = re.sub(r'#[^)]+', '', cur_sentence)
```

전처리예시: 금액 혹은 숫자, 카드사명을 언급한 텍스트에서 ()/() 꼴로 나타났으며 이를 한 단어로 정리함
(1)/(하나)의 경우 '하나로', (3개월)/(삼개월)의 경우 '삼개월'로 바꾸는 등

콜센터 데이터 최종

상담데이터수집및전처리결과

```
# 다운 샘플링 진행
```

```
counseling_전체 = down_sampling(counseling_전체)
```

다운 샘플링 된 전체 문장 개수: 10725

다운 샘플링 성공

```
counseling_전체_text = list(counseling_전체['내용'])
```

```
len_상담_text = len(counseling_전체_text)
```

```
print(f' 다운 샘플링 후 상담 데이터 건수: {len_상담_text}')
```

다운 샘플링 후 상담 데이터 건수: 200

상담데이터

200건, 전체 문장 개수 10725개

4 모델링

4-1. NER로 문장 특징 일반화



보이스피싱 / 콜센터 대화 구성

보이스피싱/콜센터데이터의 공통된 특징은
유사한 단어 무리를 자주 사용한다는 점임

'이 방, 뭐 그래서 제가 기간이 좀 걸리니까 추천 드리지 않고, 단 시간에 **신용 점수**를 상향시킬 수 있는 방법이 있는데, **채무 채무 상환 능력** 평가를 하는 방법이에요.'

'쉽게 이야기해서 뭐 이 **채무 상환 능력**은 고객님께서 뭐 신규 **대출** 1건 발생시키시고, 쓰지는 마시고 바로 일시 **상환**을 하시는 거예요.'

'그러면 고객님께서 전화를 받으시다가 일시 **상환**하신 기록이 남으시잖아요.'

'자금이 필요해서 **대출**을 받긴 하셨지만 바로 갚을 수 있는 능력이 있다고 판단되요.'

<보이스피싱데이터일부>

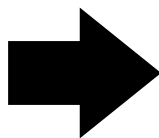
'요즘은 시대가 시대인지라 그런지 굳이 **은행** 가지 않아도 웬만한 금융 문제는 처리가 가능하니 너무 편리하네요.'

'정확하게 상담받고 가입하고 싶어서 꼭 **은행** 가서 해야 하나 걱정이 많았는데 진짜 다행입니다.'

'혹시 현재 전화로 가입 가능한 **예금 상품**이 연두 슈퍼 주거래 **예금**이랑 우리 연두 **예금**이랑 스타 연두 **예금**으로 알고 있는데 제가 정확하게 알고 있는 게 맞을까요?'

'네, 맞습니다.'

<콜센터데이터일부>

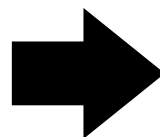


금융 도메인 분야에서 자주 사용되는 단어 및
고객의 개인 정보를 물어보는 단어가 자주 등장하는 사실을 발견함

보이스피싱 / 콜센터 대화 구성

자주 나오는 단어들을 일관성 있게 학습하고자
자연어처리의 NER 아이디어를 차용

안녕하세요 고객님? 신한은행 직원 홍길동입니다.
안녕하세요 고객님? 국민은행 직원 홍길동입니다.
안녕하세요 고객님? 우리은행 직원 홍길동입니다.



안녕하세요 고객님? <경제기관> 직원 홍길동입니다.
안녕하세요 고객님? <경제기관> 직원 홍길동입니다.
안녕하세요 고객님? <경제기관> 직원 홍길동입니다.

NER은 Named Entity Recognition으로,
특정 개체명이 언급되면 미리 정의된 분류로
바꾸며 정보를 일관성 있게 바꿔줌

<NER예시>

원래문장: 연두은행이해수 상담원입니다.



바뀐문장: <경제기관>은행이해수<직위>입니다.

원래문장: 보상할수록 십만원에서차감되는것맞죠?

바뀐문장: 보상할수록<금액>에서차감되는것맞죠?

ETRI NER 적용

ETRI NER 태그가 등장한 빈도수와
해당 태그가 보이스피싱 분류에 도움이 될지를 고민하여 반영할 태그 결정

TTA Standard	정보통신단체표준(국문표준) TTAK.KO-10.0852	제정일: 2015년 12월 16일
	개체명 태그 세트 및 태깅 말뭉치	
	Tag Set and Tagged Corpus for Named Entity Recognition	
		
	 한국정보통신기술협회 Telecommunications Technology Association	

빈도수	영문태그명	한글태그명
738	QT_PRICE	금액
555	CV_POSITION	직위
454	CV_RELATION	인간관계
144	OGG_ECONOMY	경제기관
244	CV_OCCUPATION	직업
54	OGG_POLITICS	공공기관
576	DT_DURATION	기간
348	TMI_HW	하드웨어
337	TMI_SERVICE	IT서비스
70	TI_DURATION	기간
62	CV_LAW	법률

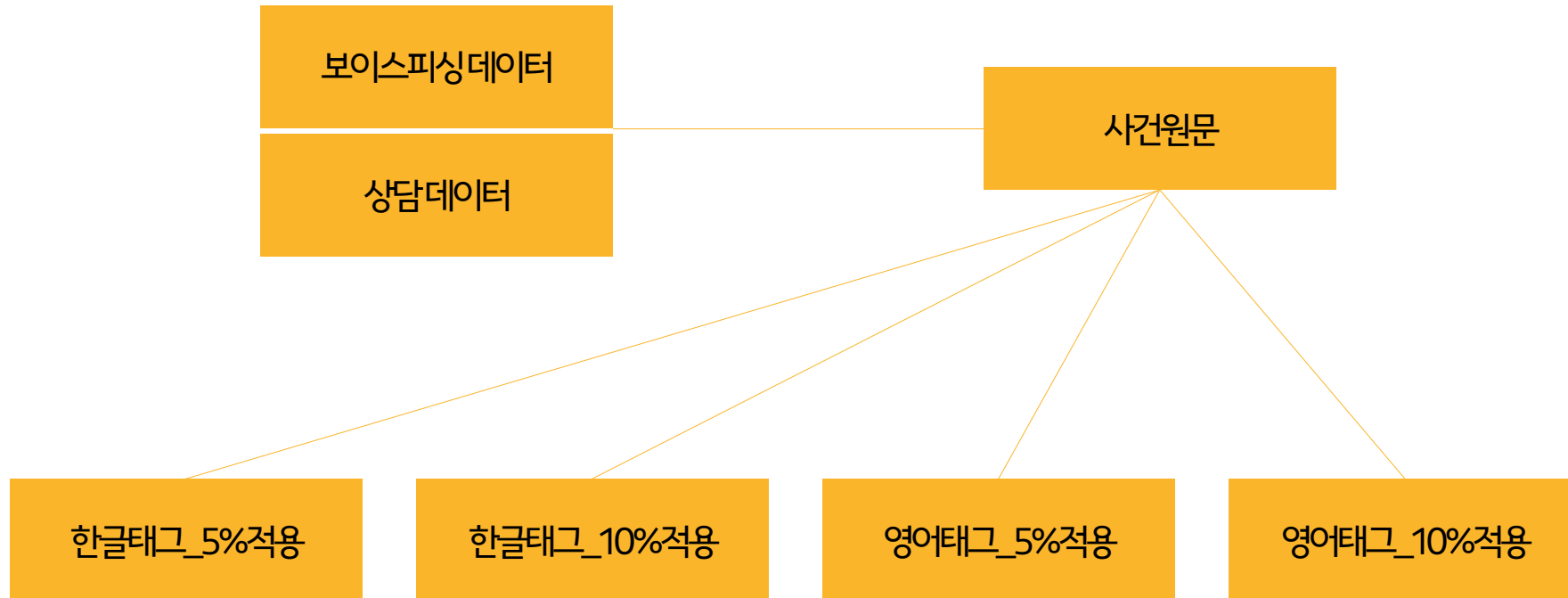
중요태그5%

중요태그10%

ETRI에서제공한개체명인식가이드라인역시참고함

ETRI NER 적용

기존의 보이스피싱/상담데이터셋에
ETRI NER 태깅을 수행하여 새로운 버전의 데이터셋 생성



*한글 기반의 사전학습 모델인 KoBERT 임베딩을 고려하여 NER 태그를 한글로 변환하여 사용 및 비교를 위한 영어 태그도 사용

ETRI NER 적용

kor_ratio5, kor_ratio10 적용 예시
(NER태깅을 한글로 중요 태그 5%, 10% 반영한 데이터)

[네, 안녕하세요., 오늘 제가 연락드린 건 제가 지금 대출을 받아서 갚고 있는 중...
[문의하고 싶은 게 있어서요., 담보대출 관련 문의 좀 드릴게요., 죄송한데 제가 ...
[안녕하세요., 국민은행 상담사 김지원입니다., 네, 안녕하세요., 무엇을 도와드릴...
[안녕하세요., 미스 은행 상담원 은가은입니다., 무엇을 도와드릴까요?, 네, 안녕...
[네 안녕하세요., 네 제가 선물옵션 투자를 좀 하고 싶어서 전화드렸습니다., 제가...
...
[여보세요., 여보세요., 000씨 혹시 맞습니까?, 네., 저는 서울 중앙 지방 ...
[여보세요., 네., 네 안녕하세요., 혹시 000 핸드폰 맞으십니까?, 어디세요...
[수고하십니다., 여기는 서울 중앙 지검이고요., 네., 저는 1,000만 범죄 수...
[어 이름은 김호철이고 태어난 곳은 전라도 광주, 올해로 사십세된 남성에 농협 은행...
[네 여보세요., 네 네., 아 아 다른 게 아니라 그 000 본인 관련된 명의 도...

<원래보이스피싱/상담통합데이터>

[네, 안녕하세요., 오늘 제가 연락드린 건 제가 지금 대출을 받아서 갚고 있는 중...
[문의하고 싶은 게 있어서요., 담보대출 관련 문의 좀 드릴게요., 죄송한데 제가 ...
[안녕하세요., <경제기관> <직업> 김지원입니다., 네, 안녕하세요., 무엇을 도...
[안녕하세요., <경제기관> <직위> 은가은입니다., 무엇을 도와드릴까요?, 네, ...
[네 안녕하세요., 네 제가 선물옵션 투자를 좀 하고 싶어서 전화드렸습니다., 제가...

...
[여보세요., 여보세요., 000씨 혹시 맞습니까?, 네., 저는 서울 중앙 지방 ...
[여보세요., 네., 네 안녕하세요., 혹시 000 핸드폰 맞으십니까?, 어디세요...
[수고하십니다., 여기는 <공공기관>이고요., 네., 저는 <금액> 범죄 수사 일 ...
[어 이름은 김호철이고 태어난 곳은 전라도 광주, 올해로 사십세된 남성에 <경제기관...
[네 여보세요., 네 네., 아 아 다른 게 아니라 그 000 본인 관련된 명의 도...

<한글로중요태그5%적용후>

[네, 안녕하세요., 오늘 제가 연락드린 건 제가 지금 대출을 받아서 갚고 있는 중...
[문의하고 싶은 게 있어서요., 담보대출 관련 문의 좀 드릴게요., 죄송한데 제가 ...
[안녕하세요., <경제기관> <직업> 김지원입니다., 네, 안녕하세요., 무엇을 도...
[안녕하세요., <경제기관> <직위> 은가은입니다., 무엇을 도와드릴까요?, 네, ...
[네 안녕하세요., 네 제가 선물옵션 투자를 좀 하고 싶어서 <하드웨어>드렸습니다....

...
[여보세요., 여보세요., 000씨 혹시 맞습니까?, 네., 저는 서울 중앙 지방 ...
[여보세요., 네., 네 안녕하세요., 혹시 000 <하드웨어> 맞으십니까?, 어...
[수고하십니다., 여기는 <공공기관>이고요., 네., 저는 <금액> 범죄 수사 일 ...
[어 이름은 김호철이고 태어난 곳은 전라도 광주, 올해로 사십세된 남성에 <경제기관...
[네 여보세요., 네 네., 아 아 다른 게 아니라 그 000 본인 관련된 명의 도...

<한글로중요태그10%적용후>

ETRI NER 적용

eng_ratio5, eng_ratio10 적용 예시
(NER태깅을 영어로 중요 태그 5%, 10% 반영한 데이터)

[네, 안녕하세요., 오늘 제가 연락드린 건 제가 지금 대출을 받아서 갚고 있는 중...
[문의하고 싶은 게 있어서요., 담보대출 관련 문의 좀 드릴게요., 죄송한데 제가 ...
[안녕하세요., 국민은행 상담사 김지원입니다., 네, 안녕하세요., 무엇을 도와드릴...
[안녕하세요., 미스 은행 상담원 은가은입니다., 무엇을 도와드릴까요?, 네, 안녕...
[네 안녕하세요., 네 제가 선물옵션 투자를 좀 하고 싶어서 전화드렸습니다., 제가...
...
[여보세요., 여보세요., 000씨 혹시 맞습니까?, 네., 저는 서울 중앙 지방 ...
[여보세요., 네., 네 안녕하세요., 혹시 000 핸드폰 맞으십니까?, 어디세요...
[수고하십니다., 여기는 서울 중앙 지검이고요., 네., 저는 1,000만 범피 수...
[어 이름은 김호철이고 태어난 곳은 전라도 광주, 올해로 사십세된 남성에 농협 은행...
[네 여보세요., 네 네., 아 아 다른 게 아니라 그 000 본인 관련된 명의 도...

<원래보이스피싱/상담통합데이터>

[네, 안녕하세요., 오늘 제가 연락드린 건 제가 지금 대출을 받아서 갚고 있는 중...
[문의하고 싶은 게 있어서요., 담보대출 관련 문의 좀 드릴게요., 죄송한데 제가 ...
[안녕하세요., <OGG_ECONOMY> <CV_OCCUPATION> 김지원입니다....
[안녕하세요., <OGG_ECONOMY> <CV_POSITION> 은가은입니다., ...
[네 안녕하세요., 네 제가 선물옵션 투자를 좀 하고 싶어서 전화드렸습니다., 제가...

...
[여보세요., 여보세요., 000씨 혹시 맞습니까?, 네., 저는 서울 중앙 지방 ...
[여보세요., 네., 네 안녕하세요., 혹시 000 핸드폰 맞으십니까?, 어디세요...
[수고하십니다., 여기는 <OGG_POLITICS>이고요., 네., 저는 <QT_P...
[어 이름은 김호철이고 태어난 곳은 전라도 광주, 올해로 사십세된 남성에 <OGG_...
[네 여보세요., 네 네., 아 아 다른 게 아니라 그 000 본인 관련된 명의 도...

<영어로 중요 태그 5% 적용 후>

[네, 안녕하세요., 오늘 제가 연락드린 건 제가 지금 대출을 받아서 갚고 있는 중...
[문의하고 싶은 게 있어서요., 담보대출 관련 문의 좀 드릴게요., 죄송한데 제가 ...
[안녕하세요., <OGG_ECONOMY> <CV_OCCUPATION> 김지원입니다....
[안녕하세요., <OGG_ECONOMY> <CV_POSITION> 은가은입니다., ...
[네 안녕하세요., 네 제가 선물옵션 투자를 좀 하고 싶어서 <TMI_HW>드렸습니...

...
[여보세요., 여보세요., 000씨 혹시 맞습니까?, 네., 저는 서울 중앙 지방 ...
[여보세요., 네., 네 안녕하세요., 혹시 000 <TMI_HW> 맞으십니까?,...
[수고하십니다., 여기는 <OGG_POLITICS>이고요., 네., 저는 <QT_P...
[어 이름은 김호철이고 태어난 곳은 전라도 광주, 올해로 사십세된 남성에 <OGG_...
[네 여보세요., 네 네., 아 아 다른 게 아니라 그 000 본인 관련된 명의 도...

<영어로 중요 태그 10% 적용 후>

4 모델링

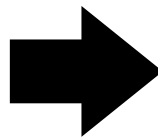
4-2. 전화 맥락을 반영한 임베딩



전화 맥락의 구성

두 사람의 전화로 이루어진 텍스트 데이터는
혼자서는 절대로 완성할 수 없는 형태며, 이 특징은 데이터에 고스란히 나타남

- 1) 보이스피싱/콜센터 데이터에서는
질문과 답변에 따른 대화 구성이 많음
- 2) 두 사람이 서로 대화하며 완성하는 텍스트로서
누군가 대화를 주도함에 따라 대화 맥락이 달라짐



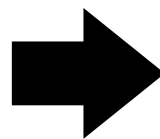
보이스피싱/콜센터 데이터 사이의 차이점
상담자 혹은 범죄자가 고객을 속일 의도를 가지는지
금품 갈취를 위한 목적성을 가지고 대화를 진행하는지



전화 맥락의 구성

주고 받는 대화 맥락을 학습하고자
자연어처리에서의 N-gram 아이디어를 차용

- 1) 보이스피싱/콜센터 데이터에서는
질문과 답변에 따른 대화 구성이 많음
- 2) 두 사람이 서로 대화하며 완성하는 텍스트로서
누군가 대화를 주도함에 따라 대화 맥락이 달라짐



보이스피싱/콜센터 데이터 사이의 차이점
상담자 혹은 범죄자가 고객을 속일 의도를 가지는지
금품 갈취를 위한 목적성을 가지고 대화를 진행하는지

<N-gram 단어뭉치 예시>

N-gram은 N개의 연속적인 나열을 의미하며,
말뭉치에서 N개의 문장 뭉치 단위로 끊어서
이를 하나의 토큰으로 간주함

원래문장: 안녕하세요 저희 팀 이름은 그놈목소리입니다.

2gram 단어뭉치: 안녕하세요 저희, 저희 팀, 팀 이름은, 그놈목소리입니다.

3gram 단어뭉치: 안녕하세요 저희 팀, 저희 팀 이름은, 팀 이름은 그놈목소리입니다.

4gram 단어뭉치: 안녕하세요 저희 팀 이름은, 저희 팀 이름은 그놈목소리입니다.

문장 N-gram 적용

기존 N-gram의 단어뭉치를
문장 뭉치로 적용하여 데이터셋을 재구성함

```
# 입력 content를 ngram 리스트로 변환
def content2ngram(content, n):
    """
    content: 사건 내용 리스트
    n: ngram으로 자를 단위
    """
    ngram_list = [] # 전체 텍스트 ngram
    for case in content:
        cut_count = len(case)-n # ngram으로 자를 횟수
        ngram_case = [] # 사건당 ngram

        for i in range(cut_count):
            ngram_group = '' # 합쳐질 ngram
            for j in range(n):
                if j != 0: ngram_group += ' '
                ngram_group += case[i+j] # 문장 합쳐서 붙임
            ngram_case.append(ngram_group)
        ngram_list.append(ngram_case)
    return ngram_list

# 데이터 ngram 변환
for dataname in datanames:
    rawdata = globals()[f'{dataname}_content']
    for n in [2, 3, 4]: # 2gram, 3gram, 4gram 데이터 생성
        globals()[f'{dataname}_content_{n}gram'] = content2ngram(rawdata, n)
```

문장 N-gram 적용

original_content 2gram 적용 예시
(NER 태깅을 수행하지 않은 원문 텍스트에서의 2gram 적용)

original_content[1]	original_content_2gram[1]
<p>['문의하고 싶은 게 있어서요.', '담보대출 관련 문의 좀 드릴게요.', '죄송한데 제가 돈 은행을 한 번도 이용한 적 없어서요.', '그래도 확인해야 하나요?', '네 제가 대출을 하고 싶은데 자동차 담보대출을 하려고 합니다.', '가능할까요?', '네 그렇게 해주세요.', '그럼 제 신용조회를 한다는 거죠?', '네 전 김길동입니다.', '구십 년 시 월 이십 오 일이고요.', '네 눌렀습니다.', '그럼 제가 얼마나 받을 수 있는 거죠?', '저는 최대로 받고 싶은데요?', '금리도 최대한 우대받고 싶은데 제가 돈 은행과 거래가 없어서 우대금리는 못 받겠죠?', '감사합니다.', '저는 최대한 최저금리로 이용하고 싶습니다.', '지금 회사 사정이 안 좋아서요.', '차량은 제 소유가 맞습니다.', '회사 차량은 아닙니다.', '그리고 차량 외에 제 신용 등급도 혹시 담보대출에 영향을 주나요?', '그렇군요.'],</p>	<p>['문의하고 싶은 게 있어서요. 담보대출 관련 문의 좀 드릴게요.', '담보대출 관련 문의 좀 드릴게요. 죄송한데 제가 돈 은행을 한 번도 이용한 적 없어서요.', '죄송한데 제가 돈 은행을 한 번도 이용한 적 없어서요. 그래도 확인해야 하나요?', '그래도 확인해야 하나요? 네 제가 대출을 하고 싶은데 자동차 담보대출을 하려고 합니다.', '네 제가 대출을 하고 싶은데 자동차 담보대출을 하려고 합니다. 가능할까요?', '가능할까요? 네 그렇게 해주세요.', '네 그렇게 해주세요. 그럼 제 신용조회를 한다는 거죠?', '그럼 제 신용조회를 한다는 거죠? 네 전 김길동입니다.', '네 전 김길동입니다. 구십 년 시 월 이십 오 일이고요.', '구십 년 시 월 이십 오 일이고요. 네 눌렀습니다.', '네 눌렀습니다. 그럼 제가 얼마나 받을 수 있는 거죠?', '그럼 제가 얼마나 받을 수 있는 거죠? 저는 최대로 받고 싶은데요?', '저는 최대로 받고 싶은데요? 금리도 최대한 우대받고 싶은데 제가 돈 은행과 거래가 없어서 우대금리는 못 받겠죠?', '금리도 최대한 우대받고 싶은데 제가 돈 은행과 거래가 없어서 우대금리는 못 받겠죠? 감사합니다.', '감사합니다. 저는 최대한 최저금리로 이용하고 싶습니다.', '저는 최대한 최저금리로 이용하고 싶습니다. 지금 회사 사정이 안 좋아서요.', '지금 회사 사정이 안 좋아서요. 차량은 제 소유가 맞습니다.', '차량은 제 소유가 맞습니다. 회사 차량은 아닙니다.', '회사 차량은 아닙니다. 그리고 차량 외에 제 신용 등급도 혹시 담보대출에 영향을 주나요?', '그리고 차량 외에 제 신용 등급도 혹시 담보대출에 영향을 주나요? 그렇군요.', '그렇군요. 그럼 집안에 있는 다른 차량도 같이 대출 신청할 수 있을까요?'],</p>

<2gram 적용 전>

<2gram 적용 후>

문장 N-gram 적용

kor_ratio5_content 3gram 적용예시
(중요태그의 5%를 한글로 NER 태깅을 수행한 텍스트에서의 3gram 적용)

kor_ratio5_content[10]

['안녕하세요.',
'<경제기관> 은행 이해수 <직위>입니다.',
'네, 안녕하세요.',
'다름이 아니라 제가 은행에 방문할 시간이 없어서 직접 가서 상담이나 신청은 못 받고 전화로도
드렸어요.',
'네, 가능합니다.',
'제가 인터넷으로 좀 알아보긴 했는데 입출식 계좌를 보유하고 있지만 하면 인터넷뱅킹 금융상품
통화 통해서 원하는 예금 상품에 가입할 수 있다고 알고 있거든요.',
'이 내용이 맞을까요?',
'네, 맞습니다.',
'요즘은 시대가 시대인지라 그런지 굳이 은행 가지 않아도 웬만한 금융 문제는 처리가 가능하
'정확하게 상담받고 가입하고 싶어서 꼭 은행 가서 해야 하나 걱정이 많았는데 진짜 다행입니다
'혹시 현재 전화로 가입 가능한 예금 상품이 <경제기관> 슈퍼 주거래 예금이란 우리 <경제기관>
는데 제가 정확하게 알고 있는 게 맞을까요?',
'네, 맞습니다.',
'근데 정기예금으로 하면 좋을 것 같은데 정기예금이라는 게 정해진 기간 동안만 정해진 금액만
'보니까 또 잘 안 되는 부분이 있더라고요.',

<3gram 적용전>

kor_ratio5_content_3gram[10]

['안녕하세요. <경제기관> 은행 이해수 <직위>입니다. 네, 안녕하세요.',
'<경제기관> 은행 이해수 <직위>입니다. 네, 안녕하세요. 다름이 아니라 제가 온
전화로도 예금 상담이 가능한가 해서 문의드리려고 연락드렸어요.',
'네, 안녕하세요. 다름이 아니라 제가 은행에 방문할 시간이 없어서 직접 가서
문의드리려고 연락드렸어요. 네, 가능합니다.',
'다름이 아니라 제가 은행에 방문할 시간이 없어서 직접 가서 상담이나 신청은
드렸어요. 네, 가능합니다. 제가 인터넷으로 좀 알아보긴 했는데 입출식 계좌를
한 후에 <직위>이랑 전화 통화 통해서 원하는 예금 상품에 가입할 수 있다고 알고
'네, 가능합니다. 제가 인터넷으로 좀 알아보긴 했는데 입출식 계좌를 보유하고
직위>이랑 전화 통화 통해서 원하는 예금 상품에 가입할 수 있다고 알고 있거든요
'제가 인터넷으로 좀 알아보긴 했는데 입출식 계좌를 보유하고 있지만 하면 인터
통화 통해서 원하는 예금 상품에 가입할 수 있다고 알고 있거든요. 이 내용이 맞
'이 내용이 맞을까요? 네, 맞습니다. 요즘은 시대가 시대인지라 그런지 굳이 은행
하네요.',
'네, 맞습니다. 요즘은 시대가 시대인지라 그런지 굳이 은행 가지 않아도 웬만한
받고 가입하고 싶어서 꼭 은행 가서 해야 하나 걱정이 많았는데 진짜 다행입니다
'요즘은 시대가 시대인지라 그런지 굳이 은행 가지 않아도 웬만한 금융 문제는
싶어서 꼭 은행 가서 해야 하나 걱정이 많았는데 진짜 다행입니다. 혹시 현재 전
우리 <경제기관> 예금이란 <직위> <경제기관> 예금으로 알고 있는데 제가 정확하
'정확하게 상담받고 가입하고 싶어서 꼭 은행 가서 해야 하나 걱정이 많았는데
<경제기관> 슈퍼 주거래 예금이란 우리 <경제기관> 예금이란 <직위> <경제기관> 예
맞습니다.',
'혹시 현재 전화로 가입 가능한 예금 상품이 <경제기관> 슈퍼 주거래 예금이란
는데 제가 정확하게 알고 있는 게 맞을까요? 네, 맞습니다. 근데 정기예금으로 하
진 금액만 예치해 두는 상품입니다.',
'네, 맞습니다. 근데 정기예금으로 하면 좋을 것 같은데 정기예금이라는 게 정하
또 잘 안 되는 부분이 있더라고요.',
'근데 정기예금으로 하면 좋을 것 같은데 정기예금이라는 게 정해진 기간 동안만
부분이 있더라고요. 게다가 정기예금 금리도 알아보니까 가장 높은 금리도 영 점
래도 적금보다는 부담이 적으니까 하는 게 좋을까요?',
'보니까 또 잘 안 되는 부분이 있더라고요. 게다가 정기예금 금리도 알아보니까

<3gram 적용후>

임베딩 진행

문장의 표현력을 극대화하고자
텍스트를 768차원으로 임베딩하는 Sentence Transformer를 선택했으며
임베딩 과정에서는 한국어에서도 좋은 성능을 보이는 KoBERT를 이용하였음

```
class SentenceTagger:
    def __init__(self):
        self.device = torch.device('cuda')
        self.embedding_model = None

    # SentenceTransformer 모델 로드
    def set_model(self):
        self.embedding_model = SentenceTransformer('./KoSentenceBERT_SKTBERT/output/training_con',
                                                    device=self.device)

    # 전체에서 사건별 임베딩
    def sentence_embedding(self, content):
        content_embedding = [] # 전체 임베딩
        for case in tqdm(content):
            case_embedding = [] # 사건별 임베딩
            for sent in case:
                sent_embedding = self.embedding_model.encode(sent, device=self.device) # 문장별 임베딩
                case_embedding.append(np.array(sent_embedding, dtype=object))
            content_embedding.append(np.array(case_embedding, dtype=object))
        return np.array(content_embedding, dtype=object)
```

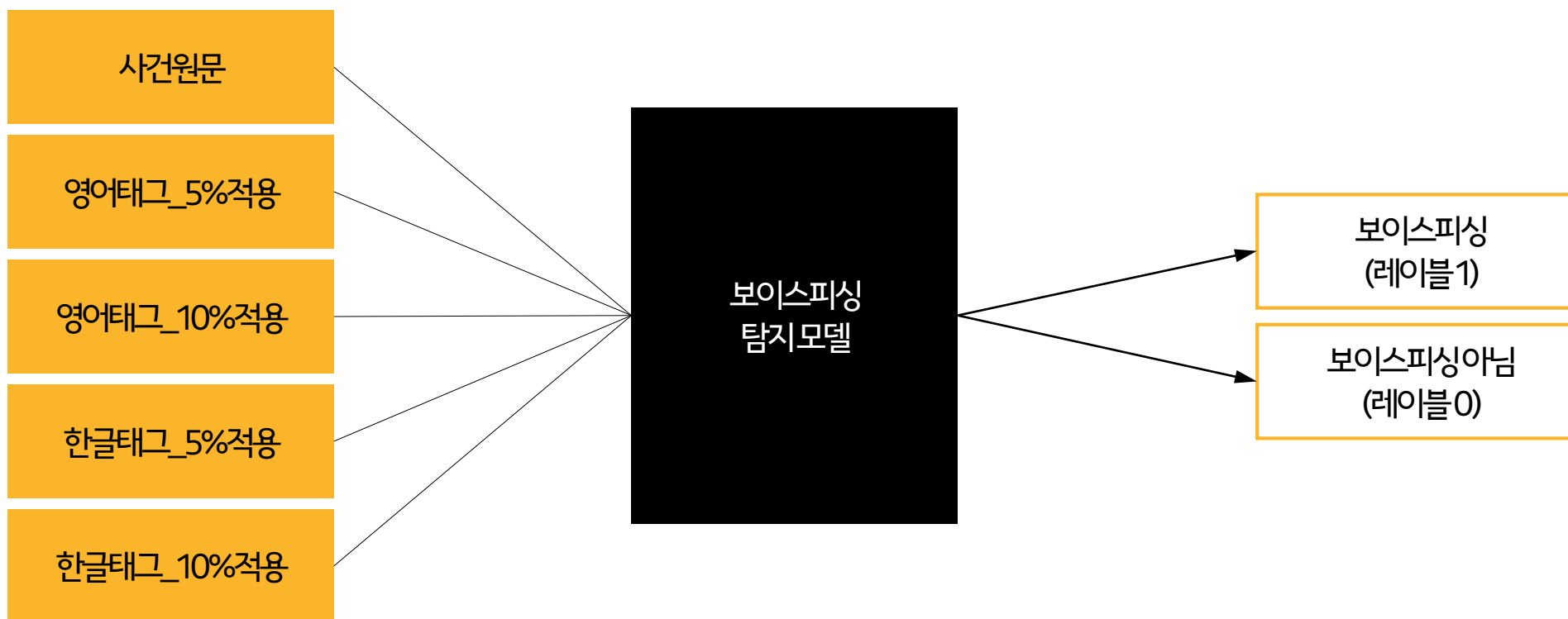
4 모델링

4-3. 보이스피싱 탐지 모델



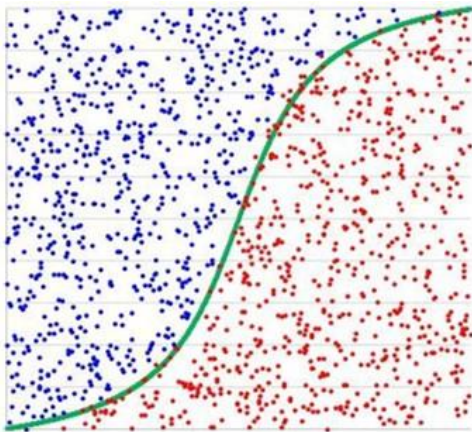
보이스피싱 탐지 모델

비교적 적은 수의 데이터에도 높은 적합성과 예측력을 보이고자 머신러닝 분류모델을 활용함

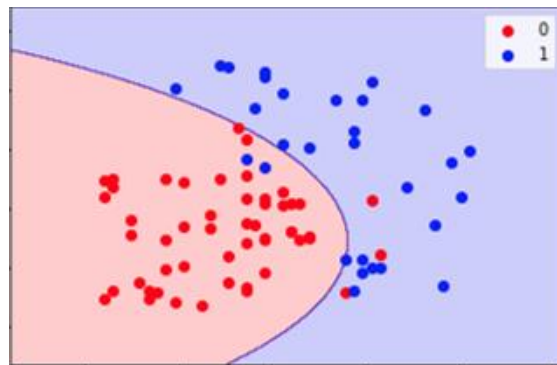


보이스피싱 탐지 모델

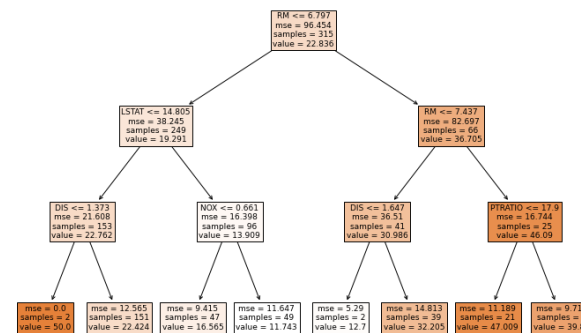
비교적 적은 수의 데이터에도 높은 적합성과 예측력을 보이고자 머신러닝 분류모델을 활용함
모델 과적합을 막기 위해 K-Fold 교차검증을 수행하여 모델별 평균 스코어값을 확인하였음



<로지스틱 회귀모델>



<나이브베이지 모델>



<랜덤포레스트 모델>

성능 평가: 모델 측면

	로지스틱 회귀모델				나이브 베이즈 모델				랜덤포레스트 모델			
Embedded Data	정확도	F1-score	재현율	정밀도	정확도	F1-score	재현율	정밀도	정확도	F1-score	재현율	정밀도
원문_2gram	0.9259	0.9268	0.9255	0.9280	0.7467	0.7400	0.7117	0.7708	0.8702	0.8727	0.8782	0.8673
원문_3gram	0.9585	0.9592	0.9603	0.9582	0.7814	0.7776	0.7521	0.8051	0.9014	0.9041	0.9143	0.8941
원문_4gram	0.9751	0.9756	0.9751	0.9760	0.8123	0.8111	0.7902	0.8332	0.9287	0.9310	0.9436	0.9187
영어태그_5%_2gram	0.9225	0.9235	0.9238	0.9233	0.7534	0.7515	0.7362	0.7676	0.8663	0.8684	0.8706	0.8662
영어태그_5%_3gram	0.9520	0.9528	0.9532	0.9524	0.7918	0.7920	0.7801	0.8044	0.8948	0.8970	0.9017	0.8925
영어태그_5%_4gram	0.9710	0.9715	0.9724	0.9707	0.8115	0.8130	0.8036	0.8228	0.9177	0.9198	0.9256	0.9141
영어태그_10%_2gram	0.9206	0.9218	0.9228	0.9208	0.7503	0.7504	0.7412	0.7600	0.8638	0.8663	0.8714	0.8614
영어태그_10%_3gram	0.9527	0.9535	0.9532	0.9538	0.7875	0.7892	0.7828	0.7958	0.8923	0.8946	0.8998	0.8896
영어태그_10%_4gram	0.9694	0.9700	0.9691	0.9708	0.8075	0.8101	0.8053	0.8151	0.9159	0.9181	0.9252	0.9112
한글태그_5%_2gram	0.9241	0.9251	0.9237	0.9265	0.7634	0.7589	0.7350	0.7844	0.8677	0.8703	0.8758	0.8650
한글태그_5%_3gram	0.9545	0.9552	0.9544	0.9560	0.7955	0.7939	0.7747	0.8140	0.8987	0.9011	0.9086	0.8938
한글태그_5%_4gram	0.9728	0.9733	0.9728	0.9738	0.8241	0.8240	0.8081	0.8408	0.9225	0.9246	0.9327	0.9166
한글태그_10%_2gram	0.9221	0.9231	0.9218	0.9244	0.7627	0.7587	0.7362	0.7827	0.8648	0.8677	0.8754	0.8603
한글태그_10%_3gram	0.9548	0.9555	0.9555	0.9555	0.7945	0.7931	0.7751	0.8120	0.8993	0.9017	0.9098	0.8939
한글태그_10%_4gram	0.9717	0.9722	0.9710	0.9734	0.8232	0.8236	0.8098	0.8380	0.9203	0.9225	0.9312	0.9141

최종적으로 모든 데이터셋에서 높은 score를 갖는
로지스틱 회귀모델을 사용해 보이스피싱 탐지 모델을 구축하기로 결정

성능 평가: NER 태깅 측면

	로지스틱 회귀모델			
data	정확도	F1-score	재현율	정밀도
원문_2gram	0.9259	0.9268	0.9255	0.9280
원문_3gram	0.9585	0.9592	0.9603	0.9582
원문_4gram	0.9751	0.9756	0.9751	0.9760
영어태그_5%_2gram	0.9225	0.9235	0.9238	0.9233
영어태그_5%_3gram	0.9520	0.9528	0.9532	0.9524
영어태그_5%_4gram	0.9710	0.9715	0.9724	0.9707
영어태그_10%_2gram	0.9206	0.9218	0.9228	0.9208
영어태그_10%_3gram	0.9527	0.9535	0.9532	0.9538
영어태그_10%_4gram	0.9694	0.9700	0.9691	0.9708
한글태그_5%_2gram	0.9241	0.9251	0.9237	0.9265
한글태그_5%_3gram	0.9545	0.9552	0.9544	0.9560
한글태그_5%_4gram	0.9728	0.9733	0.9728	0.9738
한글태그_10%_2gram	0.9221	0.9231	0.9218	0.9244
한글태그_10%_3gram	0.9548	0.9555	0.9555	0.9555
한글태그_10%_4gram	0.9717	0.9722	0.9710	0.9734

동일한 N-gram에서 NER 태깅에 따른 성능 평가

- 1) 원문을 학습한 모델과 NER 태그 적용 모델 모두 우수한 성능을 보임
- 2) 현재 모델의 경우는 학습 데이터에 맞춰 최적화된 것으로 판단됨
- 3) 추후에 변형된 범죄 유형이 나타날 경우,
NER 태그 적용 모델이 그에 따라 더욱 강인(robust)할 것으로 기대됨

성능 평가: N-gram 측면

	로지스틱 회귀모델			
data	정확도	F1-score	재현율	정밀도
원문_2gram	0.9259	0.9268	0.9255	0.9280
원문_3gram	0.9585	0.9592	0.9603	0.9582
원문_4gram	0.9751	0.9756	0.9751	0.9760
영어태그_5%_2gram	0.9225	0.9235	0.9238	0.9233
영어태그_5%_3gram	0.9520	0.9528	0.9532	0.9524
영어태그_5%_4gram	0.9710	0.9715	0.9724	0.9707
영어태그_10%_2gram	0.9206	0.9218	0.9228	0.9208
영어태그_10%_3gram	0.9527	0.9535	0.9532	0.9538
영어태그_10%_4gram	0.9694	0.9700	0.9691	0.9708
한글태그_5%_2gram	0.9241	0.9251	0.9237	0.9265
한글태그_5%_3gram	0.9545	0.9552	0.9544	0.9560
한글태그_5%_4gram	0.9728	0.9733	0.9728	0.9738
한글태그_10%_2gram	0.9221	0.9231	0.9218	0.9244
한글태그_10%_3gram	0.9548	0.9555	0.9555	0.9555
한글태그_10%_4gram	0.9717	0.9722	0.9710	0.9734

동일한NER태깅에서 N-gram에 따른 성능 평가

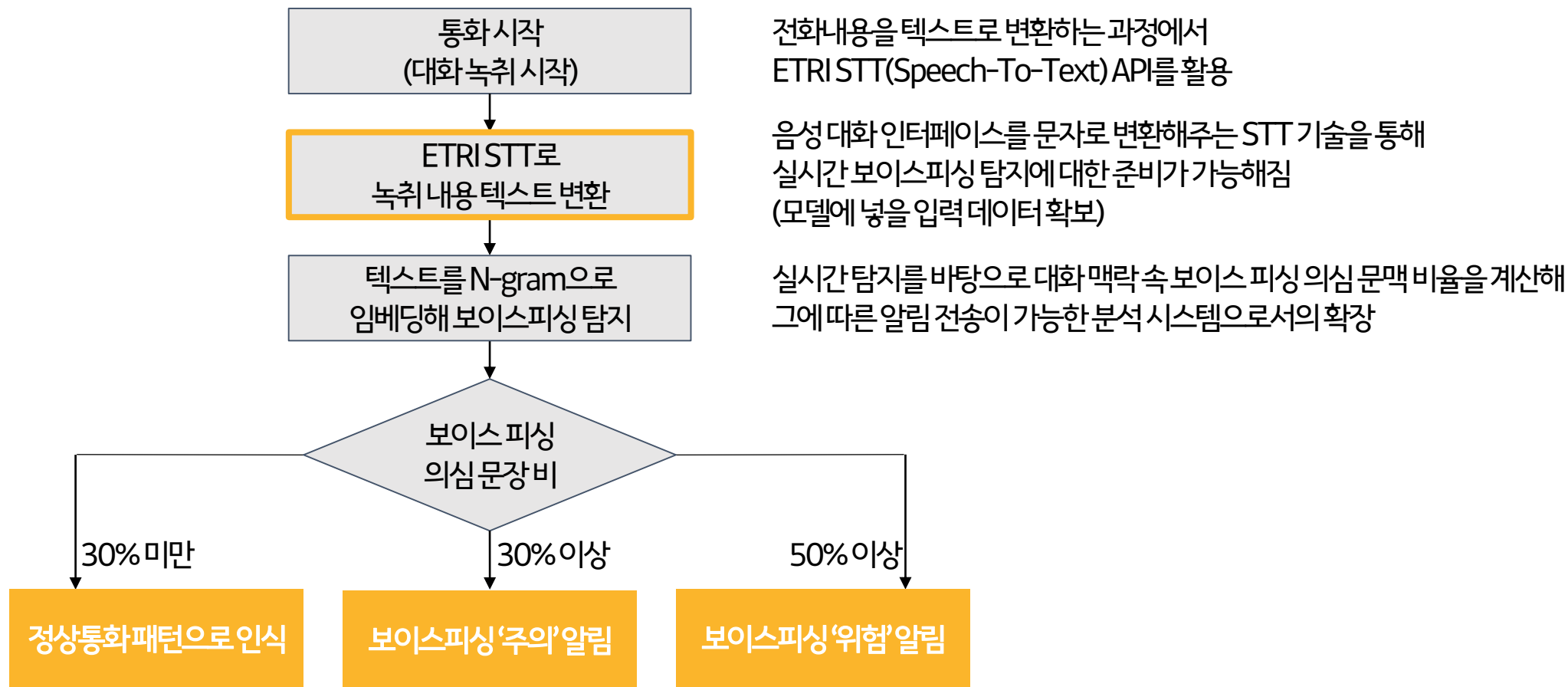
- 1) 모든 데이터셋에서 공통적으로
문장 N-gram 단위를 증가시킬수록 성능이 점진적으로 증가함
- 2) 텍스트 속 문장의 양을 많이 포함할수록,
대화 상황을 더 정확히 인식할 수 있는 것으로 보임
- 3) N-gram 단위에 따른 성능 변화는 추후 연구로 더 조사할 필요성을 가짐

5 확장

5-1. 실시간 서비스로의 확장



ETRI STT 사용 실시간 서비스로의 확장



5 확장

5-2. 금융취약계층 맞춤 서비스



금융취약계층 맞춤 시스템 제공

STEP1. 실시간보이스피싱탐지

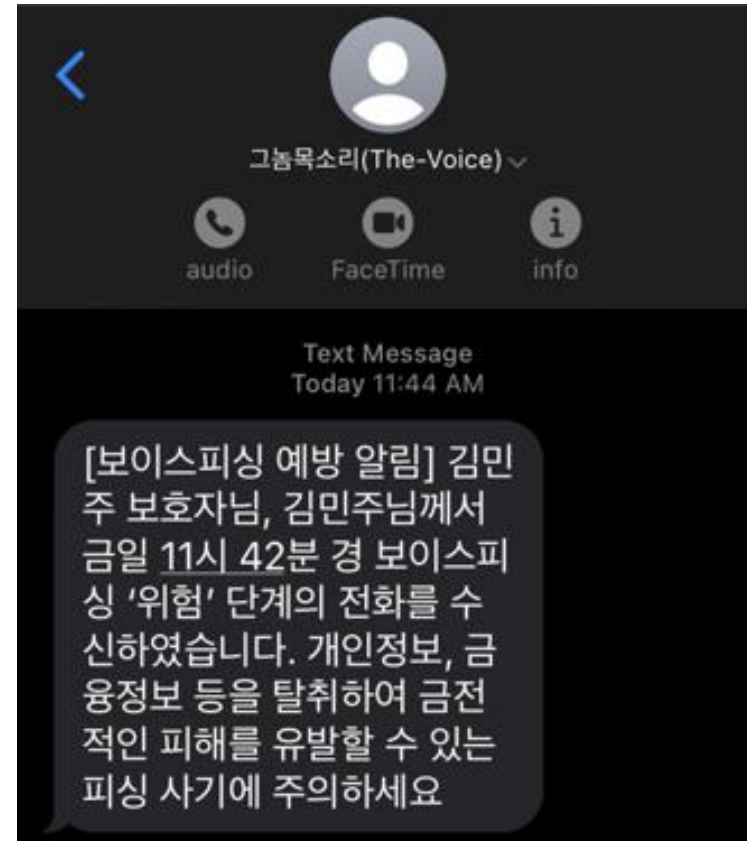
<예상시나리오>

1. 통화시작 후 1분 정도의대화가진행된이후부터
보이스피싱탐지를위한 predict 시작
2. N-gram으로 붙인데이터를 임베딩 후 예측
3. 보이스피싱이라고 예측된 문장의 비율이
일정 수준을 넘어가게 되면 그 단계에 따라 '주의', '위험' 알림 발생
 - 주의: 보이스피싱 예측 문장의 누적 비율이 30% 이상
 - 위험: 보이스피싱 예측 문장의 누적 비율이 50% 이상

STEP2. 보호자에게 위험 알림 전송

<예상시나리오>

1. 통화시작 후 1분 정도의대화가진행된이후부터
보이스피싱탐지를위한 predict 시작
2. N-gram으로 붙인데이터를 임베딩 후 예측
3. 보이스피싱이라고 예측된 문장의 비율이
50% 이상을 넘어가게 되면
저장된 보호자의 연락처로 보이스피싱 예방 알림 전송



보호자에게 알림 전송을 통해 사회적 보호막을 형성

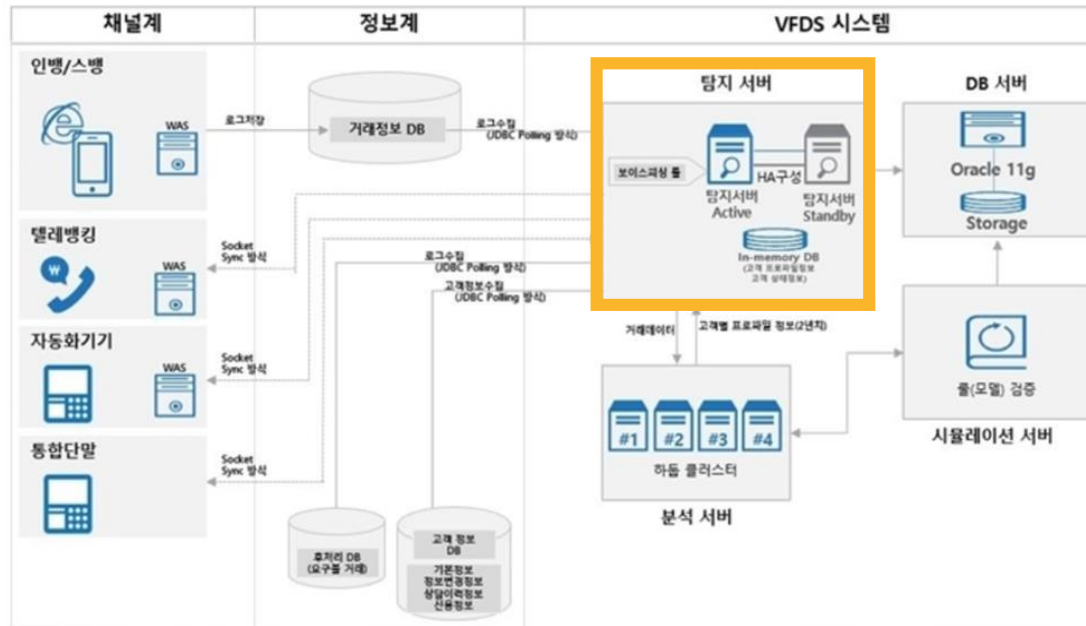
5 확장

5-3. 금융권 이상 거래 감지



금융권의 이상 거래 감지 시스템 적용

현존하는 V-FDS (보이스피싱 이상 거래 탐지 시스템)의
rule-based 보이스피싱 탐지 서버의 고도화에 활용



FDS란 고객 금융거래 패턴과 자금 흐름 등을
실시간 분석하여 이상 거래 징후를 탐지하는 시스템

현재 개발되어있는 은행의 V-FDS의
탐지 서버에 실시간 보이스피싱 탐지 모델을 적용하여
전화 등 실시간으로 이루어지는 보이스피싱 피해를 예방