산업공학 연구실
코드블라썸

STT를 이용한 환자 간병인 매칭 솔루션 기술문서

정재연 교수님, 유준영, 이문서

목차

pipeline

STT

변수화 작업

데이터 전처리

모델링 진행

STT

speech to text

Whisper (Large_v2)

환자(보호자)와 코드블라썸 간의 전화 녹음 음성을 텍스트로 변환하여 데이터를 얻기 위해 음성인식 모델을 사용해서 텍스트 데이터를 얻었습니다.

음성이 한국어로 이루어진 MP3 파일이기에 다국어에 적합한 open ai 사의 whisper 모델을 사용했습니다.

가장 무거운 모델인 large_v2모델을 사용하였고 하나의 음성파일 2.12MB (2,231,568 바이트) 당 p100 gpu를 사용했을 경우 약 5시간 정도의 시간이 소요되며 10.4KB (10,724 바이트)의 크기의 텍스트 파일이 생성됩니다.

STT

speech to text

고요. 음. 네. 이제 두번 지원금 한 10만 원 정도. 저기 그. 그. продолж해서. 예. 또 코로나iden baked reward. 이제 LiDAR resid̂o. 그래서 그가 을 consumer shelter. 그 잔액! 네. 네. 지금 한. 4박 5일 5박 6일 이렇게 되잖아요. 왜냐하면 저것은 수술

네.ших 5일 really quick igen. 네, 감사합니다. 네, 감사 네,итесь good evening. 네. 네, 감사합니다. 네. 네, 감사 감사합니다."

	file_number	CER
0	1083835111	0.171579
1	1099281403	0.167023
2	1089284499	0.262497
3	1038926896	0.233941
4	1057952048	0.177137

전체 평균 CER: 0.20079612629305488

	file_number	CER
0	1035241425	0.021213
1	1038926896	0.000000
2	1057952048	0.088805
3	1083835111	0.000000
4	1086429395	0.000000

전체 평균 CER: 0.01669169646595476

기존에 STT를 그대로 진행했을 때 한글이 아닌 다른 언어 text로 옮겨지는 문제 발견되었고
해당 문제를 해결하기 위해 Pydub의 AudioSegment를 이용하여 3분 단위로
오디오 파일을 잘라 작은 파일 단위로 만든 후 stt모델을 작동시켰습니다.

해당 과정을 수행하기 전, CER을 측정하였을 때 약 0.2에 달하던 에러율이
해당 과정 이후 0.016으로 줄어듬을 확인하였습니다.

이를 통해 해당 문제를 해결함을 알 수 있으며, 성공적인 stt과정을 수행하였습니다.

변수화 작업

data collection

chatgpt 3.5

LLM을 사용하여 stt를 통해 얻게 된 텍스트 데이터를 변수화 시켜 머신러닝을 돌리기 용이한 작업을 수행했습니다.

chat gpt3.5를 사용하였으며, 후보군이었던 chat gpt 4o와 google의 gemini를 이용하여 실험을 진행하였으나, 비용적인 문제를 비롯하여 텍스트 자체의 유의미한 변수륿 추출해내는 것에 chat gpt 3.5를 사용하는 것이 큰 품질저하를 일으키지 않아 해당 모델을 선택했습니다.

api를 불러와서 직접 노트북에 프롬프트를 입력하고 정보를 받아내는 방법으로
변수 추출을 진행하여 자동화 시스템을 구축했습니다.

해당 작업에서 프롬프트 디자인을 통해 해당 모델 api를 사용해 변수화 시킨 결과 통화 자체에 기타 변수를 고려하고 통화가 이루어지지 않았기에 많은 유의미하고 일정한 변수를 추출할 수는 없었으나 일반적으로
|[병원이름, 환자 이름, 나이, 병명] 에 대한 유의미한 정보를 습득할 수 있었습니다.

데이터 전처리

Data preprocessing

결합 및 인코딩

기존에 가지고 있던 "코드블라썸"사의 간병인 데이터와 통화를 통해 추출한 유의미한 데이터를 결합하는 과정을 진행했습니다.

[성명,간병인, 번호경력(개월,)소속, 가입일, 근무 상태, 모집채널, 성별, 국적, 나이, 생년월일, 지역1, 간병경험, 보유자격증1, 보유자격증2, 보유자격증3, 종교, 최종학력, 옷사이즈, 코로나백신선호도1, 선호도2, 선호도3, 선호도4, 선호도5, 선호도6, 선호도7, 선호도8, 선호기간, 선호병원, 기피도1, 기피도2, 기피도3, 기피도4, 기피도5, 기피도6, 기피기간, 기피병원1]이라는 특성을 보유한 간병인 데이터와

앞서 언급한 환자 데이터를 토대로 [간병 번호병원ID매니저만족도경력(개월)근무 상태성별.1국적나이지역1선호병원기피병원자격증_간병인자격증자격증_간호조무사자격증_노인심리상담사자격증_실버케어지도사자격증_요양보호사자격증_호스피스자격증_활동보조인선호_감염성질환선호_경증환자선호_고액환자선호_남성환자선호_단기환자선호_석션선호_섬망환자선호_여성환자선호_와상환자선호_장기환자선호_...]등의 특성을 지닌 환자-간병인 결합 데이터를 생성했습니다.

데이터 전처리

Data preprocessing

```
binary = ['코로나', 'CRE', 'VRE', '결핵', '환자위생관리', '환자식사보조', '환자화장실 도움',
       '환자거동 도움', '피딩', '석션', '욕창', '재활치료', '장루', '투석', '치매', '섬망', '전신마비',
       '하반신마비', '편마비', '기타증상', '요구사항']
non_bin = ['병원 이름', '환자 이름', '성별' ,'나이', '몸무게',
           '병명(진단명)', '환자의식상태', '선호간병인성별',
           '선호간병인국적', '선호간병인연령', '단축/연장 가능성']
for column in binary:
    data[column] = data[column].apply(lambda x: 0 if x == "'blank'," else 1)
for column in non_bin:
    data[column] = data[column].apply(lambda x: 0 if x == "'blank'," else x)
```

```
non_bin_data = data[non_bin]
encoder = OneHotEncoder(sparse=False, handle_unknown='ignore')
non_bin_encoded = encoder.fit_transform(non_bin_data)


non_bin_encoded_df = pd.DataFrame(non_bin_encoded, columns=encoder.get_feature_names_out(non_bin))


data = data.drop(non_bin, axis=1)
data = pd.concat([data, non_bin_encoded_df], axis=1)
data.to_csv('data/patient.csv',index=True, encoding='cp949')
```

['전화번호', '간병 시작일', '단축 가능성', '연장 가능성', '단축/연장 가능성', '병원 이름', '병동', '층', , '호실', '환자 이름', '성별', '나이', '몸무게', '병명(진단명)', '병실 유형', '입원 경로', '코로나', , 'CRE', 'VRE', '결핵', '환자의식상태', '환자위생관리', '환자식사보조', '환자화장실 도움', '환자거동 도움', '피딩', '석션', '욕창', '재활치료', '장루', '투석', '치매', '섬망', '전신마비', '하반신마비', '편마비', '기타증상', '요구사항', '선호간병인성별', '선호간병인국적', '선호간병인연령'] 의 특성 중 ["간병 시작일", '연장 가능성',"단축 가능성", '병동', '층', '호실', '병실 유형', '입원 경로']의 특성의 경우 전체 튜플들의 90%이상이 비어있는 특성이기에 제거하였습니다.

기타 이진값을 가지는 특성과 그렇지 않은 특성을 나누어 범주화 되어 있는 특성에 한해서 sklearn.preprocessing 라이브러리의 OneHotEncoder를 사용하여 인코딩을 진행했습니다.

또한, 기존의 환자 데이터에서 비어있는 상당수의 특성을 제거하고 모든 dataframe을 결합하여 utf-8-sig의 방식으로 csv화 만들었고 해당 모든 전처리를 완료했습니다.

데이터 전처리

Data preprocessing

또한, 기존의 환자 데이터에서 비어있는 상당수의 특성을 제거하고 모든 dataframe을 결합하여 utf-8-sig의 방식으로 csv화 만들었고 해당 모든 전처리를 완료했습니다.

[간병 번호, 병원ID, 매니저만족도, 경력(개월), 근무 상태, 성별, 국적, 나이, 지역1, 선호병원, 기피병원자격증_간병인자격증자격증_간호조무사자격증_노인심리상담사자격증_실버케어지도사자격증_요양보호사자격증_호스피스자격증_활동보조인선호_감염성질환선호_경증환자선호_고액환자선호_남성환자선호_단기환자선호_석션선호_섬망환자선호_여성환자선호_와상환자선호_장기환자선호_장루선호_정형외과선호_중증환자선호_치매환자선호_편마비환자선호_피딩기피_격리환자기피_경증환자기피_경추환자기피_과체중환자기피_남성환자기피_단기환자기피_석션환자기피_섬망환자기피_수면장애환자기피_암환자기피_잠못자는환자기피_장루기피_중증환자기피_치매환자기피_파킨슨병기피_폐질환기피_피딩전화번호단축/연장 가능성병원 이름환자 이름성별나이몸무게병명(진단명)코로나CREVRE결핵환자의식상태환자위생관리환자식사보조환자화장실 도움, 환자거동 도움, 피딩, 석션, 욕창, 재활치료, 장루투석, 치매, 섬망, 전신마비, 하반신마비, 편마비, 기타증상, 요구사항, 선호간병인, 성별선호간병인, 국적, 선호간병인, 연령]의 특성을 지니며 5%이하의 결측치를 보이는 csv형태의 데이터를 최종적으로 선별하게 되었습니다.

모델링 진행

modelling

```
X = relevant_columns.drop('만족도', axis = 1)
y = relevant_columns['만족도']
```

```
y_pred = RF_model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
```

최종적으로 나오게 된 데이터를 사용하여 ['경력(개월)', '성별.1', '만족도', '자격증_간병인자격증', '자격증_간호조무사', '자격증_노인심리상담사', '자격증_실버케어지도사','자격증_요양보호사', '자격증_호스피스', '자격증_활동보조인','선호_감염성질환', '선호_경증환자', '선호_고액환자', '선호_남성환자', '선호_단기환자', '선호_석션', '선호_섬망환자', '선호_여성환자', '선호_와상환자', '선호_장기환자','선호_장루', '선호_정형외과', '선호_중증환자', '선호_치매환자', '선호_편마비환자', '선호_피딩', '기피_격리환자', '기피_경증환자', '기피_경추환자', '기피_과체중환자', '기피_남성환자', '기피_단기환자', '기피_석션환자', '기피_섬망환자', '기피_수면장애환자', '기피_암환자','기피_잠못자는환자', '기피_장루', '기피_중증환자', '기피_치매환자', '기피_파킨슨병', '기피_폐질환', '기피_피딩', '치매', '섬망', '전신마비', '하반신마비', '편마비', '기타증상', '요구사항'] 중 '만족도'라는 특성을 목표 변수로 설정하였습니다.

이후, sklearn.ensemble 라이브러리를 사용하여 RandomForestRegressor라는 모델을 사용해 머신러닝 과제를 수행하였고 만족도를 추출할 수 있도록 하여 최종적인 솔루션을 마무리했습니다.

문제 인식

pipeline

STT

open ai의 whisper 모델로 mp4파일을 텍스트 파일로 전환
3분 단위로 오디오를 분할하여 모델에 삽입하면서 CER의 대폭 감소(0.2->0.016)

변수화 작업

open ai의 LLM인 chat gpt 3.5 를 사용하여 텍스트에서 유의미한 특성을 추출

데이터 결합 및 전처리

기존 간병인 데이터와 환자데이터를 결합하고 통화를 통해 얻은 유의미한 변수를
결합하여 최종 데이터셋 생성

모델링 진행

randomforest 및 기타 모델들을 사용하여 기존 데이터 셋의 특성 중 "만족도"를
목표 변수로 설정하여 머신러닝을 수행함으로 최종 솔수션 완성

감사합니다!

Thank You

가톨릭대 산업공학연구실