KIPS Trans. Comp. and Comm. Sys.
Vol.9, No.11 pp.281~290
ISSN: 2287-5891 (Print), ISSN 2734-049X (Online)


https://doi.org/10.3745/KTCCS.2020.9.11.281

# Natural Language Processing Model for Data Visualization Interaction in Chatbot Environment

Sang Heon Oh† · Su Jin Hur†† · Sung-Hee Kim†††

## ABSTRACT

With the spread of smartphones, services that want to use personalized data are increasing. In particular, healthcare-related services deal with a variety of data, and data visualization techniques are used to effectively show this. As data visualization techniques are used, interactions in visualization are also naturally emphasized. In the PC environment, since the interaction for data visualization is performed with a mouse, various filtering for data is provided. On the other hand, in the case of interaction in a mobile environment, the screen size is small and it is difficult to recognize whether or not the interaction is possible, so that only limited visualization provided by the app can be provided through a button touch method. In order to overcome the limitation of interaction in such a mobile environment, we intend to enable data visualization interactions through conversations with chatbots so that users can check individual data through various visualizations. To do this, it is necessary to convert the user's query into a query and retrieve the result data through the converted query in the database that is storing data periodically. There are many studies currently being done to convert natural language into queries, but research on converting user queries into queries based on visualization has not been done yet. Therefore, in this paper, we will focus on query generation in a situation where a data visualization technique has been determined in advance. Supported interactions are filtering on task x-axis values and comparison between two groups. The test scenario utilized data on the number of steps, and filtering for the x-axis period was shown as a bar graph, and a comparison between the two groups was shown as a line graph. In order to develop a natural language processing model that can receive requested information through visualization, about 15,800 training data were collected through a survey of 1,000 people. As a result of algorithm development and performance evaluation, about 89% accuracy in classification model and 99% accuracy in query generation model was obtained.

Keywords : Chatbot, Data Visualization, Interaction, Natural Language Processing, SQL(Structured Query Language)

# 챗봇 환경에서 데이터 시각화 인터랙션을 위한 자연어처리 모델

오 상 헌† · 허 수 진†† · 김 성 희†††

## 요 약

스마트폰의 보급으로 인해 개인화된 데이터를 활용하고자 하는 서비스들이 증가하고 있다. 특히, 헬스케어와 관련된 서비스들은 다양한 데이터를 다루며, 이를 효과적으로 보여주기 위해 데이터 시각화 기법을 활용하고 있다. 데이터 시각화 기법이 활용되면서 자연스럽게 시각화에서의 인터랙션 또한 함께 강조되고 있다. PC 환경에서 데이터 시각화에 대한 인터랙션은 마우스로 이루어지기 때문에, 데이터에 대한 필터링이 다양하게 제공되고 있다. 반면, 모바일 환경에서의 인터랙션은 화면의 크기가 작고, 인터랙션 가능 여부를 인지하기 어려워 버튼 터치 방식으로 앱에서 제공하는 제한된 시각화만을 제공받을 수 있다. 이러한 모바일 환경에서의 인터랙션 한계를 극복하기 위해, 챗봇과의 대화를 통해 데이터 시각화 인터랙션을 가능하게 하여 사용자들에게 개개인의 데이터를 다양한 시각화를 통해 확인할 수 있도록 하고자 한다. 이를 위해서는 사용자의 질의를 쿼리로 변환하여, 주기적으로 데이터를 축적하고 있는 데이터베이스에서 변환된 쿼리를 통해 결과 데이터를 불러올 수 있어야 한다. 자연어를 쿼리로 변환하는 연구는 현재 많이 이루어지고 있지만, 시각화를 기반으로 하여 사용자의 질의를 쿼리로 변환하는 연구에 대해서는 아직 이루어지지 않았다. 따라서, 본 논문에서는 사전에 데이터 시각화 기법이 정해진 상황에서의 쿼리 생성에 초점을 맞추고자 한다. 지원하는 인터랙션은 태스크 x-축 값에 대한 필터링 및 두 그룹 간 비교이다. 테스트 시나리오는 걸음 수에 대한 데이터를 활용하였으며, x-축 기간에 대한 필터링은 바 그래프, 두 그룹간 비교는 라인 그래프로 나타내었다. 시각화를 통해 요청한 정보를 제공받을 수 있는 자연어처리 모델을 개발하기 위해 1,000명을 대상으로 한 설문조사를 통해 약 15,800개의 학습 데이터를 수집하였다. 알고리즘 개발 및 성능 평가를 진행한 결과, 분류 모델에서는 약 89%, 쿼리 생성 모델에서는 약 99% 정확도를 보였다.

키워드 : 챗봇, 데이터 시각화, 인터랙션, 자연어처리, SQL(Structured Query Language)

※ 이 논문은 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원(No. 2019 R1C1C1005508) 및 과학기술정보통신부 및 정보통신기획평가원의 GrandICT 연구지원센터 지원사업의 연구결과로 수행되었음(IITP-2020-0-01791).
† 비 회 원 : 동의대학교 IT융합학과 석사과정
†† 비 회 원 : 동의대학교 산업ICT기술공학과 학사과정
††† 정 회 원 : 동의대학교 산업ICT기술공학과 조교수
Manuscript Received : September 21, 2020
Accepted : November 3, 2020
* Corresponding Author : Sung-Hee Kim(sh.kim@deu.ac.kr)

## 1. 서 론

스마트폰의 보급이후 언제 어디에서든 인터넷을 사용함에 따라 데이터가 급격하게 생성되면서 개인화된 데이터를 저장하고 활용하는 앱들이 증가하고 있다. 하루의 운동량을 기록하는 헬스케어 서비스 또는 비용 지출을 관리하는 가계부 앱

등을 통해 개인들은 생활 속에서 다양한 데이터를 축적하고 있다[1-3]. 삼성 헬스케어, 핏빗(FitBit), 구글 핏(Google Fit)과 같은 서비스들은, 이와 같은 데이터를 효과적으로 보여주기 위해서 데이터 시각화 기법을 활용하고 있다[4, 5]. 예를 들어, 월별 체지방량 변화, 주별 칼로리 소모량, 일별 걸음 수 등의 정보를 쉽게 전달하기 위해서 바 그래프 또는 라인 그래프 등을 제공하고 있다. 하지만 이러한 데이터 시각화들은 대부분 정적인 그래프로 표현되며, 시각화에 대한 인터랙션은 버튼, 드롭다운 박스, 달력 등의 컴포넌트를 통해 최소한으로 제공된다(Fig. 1 참조).

시각화에서의 인터랙션은 특정 기준의 데이터만 그래프에 표현하는 필터링, 일부 시각화 영역을 강조하는 하이라이팅, 그리고 군집되어 있는 데이터를 확대하여 보는 기능 등 데이터를 효과적으로 보기 위해서 중요한 요소이다[6]. PC 환경에서 데이터 시각화에 대한 인터랙션은 마우스로 이루어지기 때문에 데이터에 대한 필터링이 다양하게 제공되지만, 모바일 환경에서는 화면의 크기도 제한적이며 인터랙션이 가능하다는 점을 사용자가 인지하기 어려워 터치에 맞는 버튼 형태가 주를 이룬다[7]. 그에 따라 앱에서 제공하는 시각화만 제한적으로 볼 수 있다는 단점이 있다. 예를 들어 월별, 주별, 하루에 대한 걸음 수만 보여준다면, 최근 2주에 대한 데이터나 이번 주와 저번 주 데이터에 대한 비교 등과 같이 다양한 시각화를 볼 수가 없다. 이러한 한계를 극복하기 위해, 대화형으로 챗봇 환경에서 데이터 시각화 인터랙션을 가능하게 한다면 사용자들은 다양한 시각화를 통해 개인의 데이터를 볼 수가 있다.

스마트폰의 보급 이후, 모바일 환경이 사람들의 생활에 매우 밀접하게 스며들었다. 카카오톡(Kakao talk), 라인(Line) 등을 통한 채팅 환경은 이제 익숙한 생활 습관으로 자리 잡았으며, 인공지능 기술이 발전함에 따라 자연스럽게 챗봇 분야와 자연어 처리에 관련된 다양한 연구가 이루어지고 있다[8-10]. 앞서 설명한 것과 같이, 컴포넌트로만 지원되던 인터랙션을 챗봇 환경에서 대화형으로 제공한다면, 사용자들은 자연스럽게 원하는 시각화 결과를 제공 받을 수 있을 것이다. 그러기 위해서는, 사용자의 요구사항이 담긴 질의가 자연어 처리를 통해 데이터베이스 쿼리로 변화되어, 해당 데이터를 추출한 후, 시각화가 업데이트되어야 한다. 데이터베이스에 저장되어있는 데이터를 조회하기 위해서는 프로그래밍 언어 중 하나인 SQL(Structured Query Language)을 활용해야 하는데, 최근 기계학습과 딥러닝의 발전으로 자연어 처리(Natural Language Processing)로 SQL문을 생성할 수 있는 기술이 빠르게 발전하고 있다[11-13].

기존의 연구들은 모델의 성능 향상에 초점을 맞추어 알고리즘을 개발하였다. 대표적으로 위키피디아에서 수집한 데이터 셋을 기반으로 만들어진 WikiSQL 데이터셋을 활용하여 모델을 생성하고, SELECT 문을 생성해 결과 쿼리의 정확도를 예측해 SELECT 문의 전체를 생성하는 연구가 이루어져 왔다. 하지만 데이터 시각화가 주어진 상황에서, 사용자의 질의를 SQL로 변환하는 연구는 아직 이루어지지 않았으며, 본 논문에서는 이를 위해 해당 데이터 시각화 및 인터랙션 선정, 학습 데이터 구축, 알고리즘 개발 및 성능 평가를 진행하였다.

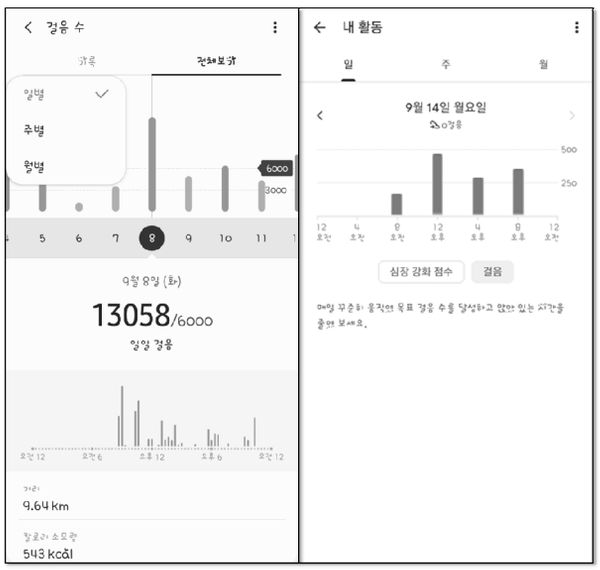



Fig. 1. Examples of Interactions Supoprted by Samsung Healthcare Apps and Google Fit Apps

본 논문의 범위는 질의를 통해 최적화된 데이터 시각화를 생성하는 것은 포함하지 않으며, 이미 데이터 시각화 기법이 정해진 상황에서의 쿼리 생성에 대해서만 다룬다. 해당 그래프는 바 그래프, 라인 그래프며, 지원하는 인터랙션 태스크는 x축에 대한 필터링, 두 개 데이터 집단에 대한 비교이다. 이는 기존의 인터랙션 방식과는 다른, 챗봇을 통한 대화 기반의 인터랙션으로, 본 논문에서는 사용자들이 원하는 정보를 직접 질문하여, 시각화를 통해 요청한 정보를 디테일하고 쉽게 제공받을 수 있도록 하는 자연어 처리 모델을 개발하였다.

## 2. 관련 연구

### 2.1 대화형 데이터 시각화 관련 연구

데이터 시각화는 빅데이터 분석에서 사용자가 데이터를 쉽게 읽고 이해할 수 있는 직관적인 방법으로, 의사결정을 위한 통합된 견해를 제시해 정책이나 서비스의 품질을 향상시키는데 도움이 된다. 또한, 시각화는 데이터 분석의 결과를 직관적으로 보여주는 것 뿐만 아니라 데이터를 수집, 정리, 분석, 공유하는 전체 과정에서도 중요한 역할을 한다. I. Ko와 H. Chang은 다양한 데이터 소스와 연결하고, 드래그 앤 드롭 인터페이스를 통해 차트, 맵, 대시보드 및 스토리를 만들어 대화형 데이터 시각화를 표현할 수 있는 Tableau를 사용하여 의료 데이터의 대화형 시각화 및 분석을 위한 절차를 제안하였다[14].

D. J. Janvrin외 2명의 연구에서는 대화형 데이터 시각화(IDV)를 위한 데이터를 구성하고, IDV가 사용자에게 미칠 수

있는 영향을 사람들이 이해할 수 있도록 하였다[15]. IDV는 복잡한 데이터를 이해하고 의사결정을 위한 도구로, 데이터 분석의 구성 요소로 사용되는 인터페이스를 통해 사용자가 데이터를 탐색, 선택 및 표시할 수 있도록 한다. 또한, 의사결정자가 표시할 데이터와 데이터를 표현하는 방법을 선택할 수 있도록 한다.

인터랙션은 정보 시각화(InfoVis)의 중요한 부분을 차지하고 있다. J. S. Yi 외 3명은 Select, Encode, Filter, and Connect 등 7가지 일반적인 인터랙션 기법을 제안하였다[16]. 이 7가지 범주는 시스템과 인터랙션을 통한 사용자의 의도를 중심으로 구성되었으며, 인터랙션 기법을 논의하고 평가하는 데 도움이 될 수 있고, 더 깊은 이해와 인터랙션의 과학을 위한 토대가 될 수 있다. 이 중 필터 인터랙션 기법은 사용자가 범위 또는 조건을 지정하여 해당 기준을 충족하는 데이터 항목만 표시되도록 하는 것이다. 이러한 인터랙션은 직관적이며 응답성이 뛰어나고, 사용자가 데이터 집합의 컨텍스트를 이해하는 데 도움을 주며, 자연스러운 시각적 탐색을 할 수 있도록 한다. 이 7가지 인터랙션 기능들은 설계자와 개발자들이 시스템에 의해 사용자의 요구가 충족되는지에 대한 여부를 검사하는 데 도움을 줄 수 있으며, 사용자는 데이터 세트에 대한 다양한 관점을 가지고 통찰력을 얻을 수 있다.

Fig. 2와 같이 데이터 시각화에서 인터랙션은 하나의 차트 안에서 다양한 기준에 따라 시각화 패턴을 확인할 수 있다. 따라서 사람들에게 자유롭게 데이터를 탐색할 기회를 제공하며, 데이터로부터 개인만의 인사이트를 찾을 수 있도록 한다.

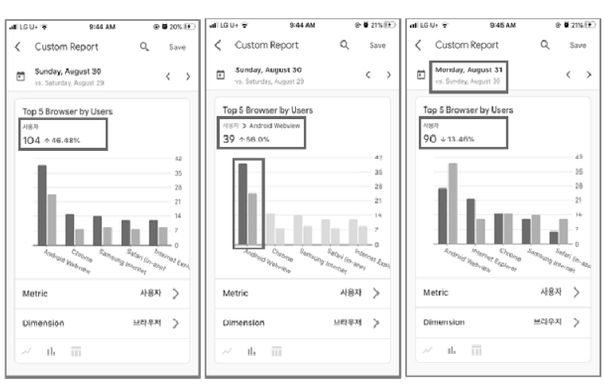



Fig. 2. Example of Data Visualization Interaction in Google Analytics

### 2.2 자연어처리 관련 연구

자연어 문제를 해당 SQL 질의로 번역하기 위한 심층신경망 Seq2SQL을 제안하는 연구가 있다[11]. 이 모델은 SQL 쿼리 구조를 활용하여 생성된 쿼리의 출력 공간을 크게 줄인다. 그리고 질의 순서가 없는 부분을 생성하기 위해 데이터베이스에 대한 루프 내 쿼리 실행의 보상을 사용하여, 교차 엔트로피 손실을 통한 최적화에 덜 적합하다는 것을 보여주었다. 또한, 모델을 훈련시키기 위해 위키백과에서 80,654개의 질문 및 SQL 질의를 수작업으로 작성한 데이터 집합 WikiSQL을 만들었다. 질의 실행 환경이 있는 정책 기반 강화 학습을 WikiSQL에 적용함으로써, Seq2SQL 모델은 실행 정확도와 논리 폼 정확도가 향상되었다.

한 연구에서는 자연어 문제를 SQL 질의에 매핑할 수 있는 생성 모델을 제시하였다[12]. 기존 단어별로 SQL 질의어를 생성하는 기존의 신경망 기반 접근 방식에서 질문과 표 내용의 불일치로 인해 부정확하고 실행 불가능한 문제를 해결하기 위해 표의 구조와 SQL 언어의 구문을 고려하였다. 가장 큰 질문-SQL 쌍을 가진 데이터 집합 WikiSQL로 연구를 진행하였으며, 컬럼 이름, 셀 또는 SQL 키워드로부터 콘텐츠를 복제하는 학습과 컬럼-셀(Column-Cell) 관계를 활용하여 WHERE 조항의 생성을 향상시켜 생성된 SQL 질의의 품질을 향상시켰다.

P. Wang 외 2명은 딥러닝 기반의 TRANslate-Edit Model for Question-to-SQL(TREQS) 생성 모델을 개발하여 의료 질문-SQL 데이터 집합의 부족 문제와 질문 용어에서의 약어와 오타로 인한 정확성에 대한 문제를 해결하였다[13]. 이 모델에서는 Sequence-to-Sequence 모델을 적용하여 해당 질문에 대한 SQL 질의를 직접 생성하며, 주의 깊게 복사하는 메커니즘과 작업별 조회표를 사용하여 필요한 편집을 추가로 수행하도록 하였다. 양적 및 정성적 실험 결과 조건 값을 예측하는 데 있어 제안된 방법의 유연성과 효율성과 약어와 오타가 있는 무작위 질문에 대한 강인성을 나타냈다.

## 3. 본 론

### 3.1 데이터 시각화 및 인터랙션 태스크

시각화를 위한 샘플 데이터는 헬스케어 관련 기존의 앱들에서 주로 사용되었으며, 스마트폰과 스마트워치 등에서 센서로 쉽게 데이터를 수집하여 활용하고 있는 걸음 수 데이터로 선정하였다(Fig. 1 참조). 시각화 방법은 기존의 타 앱에서도 주로 사용하고 있는 바 그래프와 두 데이터 집단의 비교 태스크에 적합한 라인 그래프 총 두 가지를 채택하여 시각화에 활용하도록 하였으며, 필터링과 비교 총 2개의 태스크에 대한 시각화 인터랙션을 지원하도록 한다.

### 3.2 데이터 수집

모델을 학습하기 위한 데이터는 국내 설문조사 플랫폼, 오픈 서베이를 통해 수집하였다. Fig. 3은 실제 설문에 사용한 자료로, 챗봇 환경에서 이루어질 수 있는 질문들을 수집하기 위해 샘플 데이터로 그래프를 만들었다. 이 그래프를 보기 위한 25개의 주관식 형태의 답변을 얻기 위해 1,000명을 대상으로 설문조사를 진행하였다.

쿼리 생성을 위한 학습 데이터를 새롭게 구축 해야 했기 때문에 일부 데이터 시각화와 그에 따른 인터랙션 태스크 중 특히 필터링 관련 질문의 비중을 더 크게 두고 설문 데이터를 수집하였다. Table 1의 Number는 설문조사 문항 번호를 의미하며, Contents는 해당 문항에서의 시각화 문항을 의미한다. 1~20번 문항은 필터링 태스크와 관련한 문항이며, 21

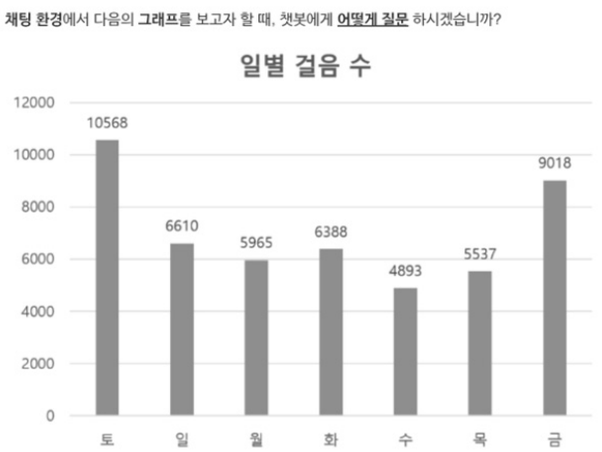



Fig. 3. Sample of a Survey Question

~25번은 비교 태스크에 관한 문항으로, 필터링 태스크에 대한 질문이 많은 부분을 차지한다.

기존의 여러 앱에서 주로 버튼이나 리스트 박스 형태로 제공되었던 일별(최근 7일), 주별(최근 5개 주), 월별(최근 5개 월) 데이터에 대한 1~3번 문항과 '최근'과 관련된 4~9번 문항, 특정 기간에 대한 10~20번 문항, 그리고 비교와 관련된 21~25번 문항으로 총 25개의 문항을 6개의 카테고리로 분류할 수 있다.

Table 1. Survey Contents

| Number | Contents |
|---|---|
| 1 | Daily step count |
| 2 | Weekly step count |
| 3 | Monthly step count |
| 4 | Step count for recent two weeks |
| 5 | This week step count |
| 6 | Step count for the past 5 days |
| 7 | Step count for this month |
| 8 | Step count for the past 3 days |
| 9 | Step count for the recent 30 days |
| 10 | Step count for last week |
| 11 | Step count for last month |
| 12 | Step count for two weeks ago |
| 13 | Step count for June |
| 14 | Step count for March |
| 15 | Step count for two months ago |
| 16 | Step count for July |
| 17 | Steps for May |
| 18 | Step count between March 6 and 14 |
| 19 | Step count between March 25 and 29 |
| 20 | Step count between June 9 and 17 |
| 21 | Comparison between this week and last week |
| 22 | Comparison between this month and last month |
| 23 | Comparison between last week and last across weeks before |
| 24 | Comparison between May and July |
| 25 | Comparison between March and May |

### 3.3 1차 분류 모델

1) 데이터 전처리

Fig. 4는 분류 모델의 전체적인 프로세스를 나타낸다. 수집된 25,000개의 데이터 중 무의미한 데이터를 필터링하여 약 15,800개의 데이터를 수집하였으며, 수집된 데이터를 학습을 위해 필터링한 결과, 데이터셋은 다음 Table 2와 같다. Sentence는 사람들이 설문에 응답한 내용이며, Label은 일별/주별/월별/최근/특정 기간/비교, 총 6개의 카테고리에 따라 각 1~6의 숫자로 라벨링한 것을 의미한다. 이를 통해 데이터가 모두 섞여 있음을 확인할 수 있다. 필터링 작업 후, 약 15,800개의 데이터 중 11,000개의 데이터를 훈련 데이터, 나머지 4,700개의 데이터를 테스트 데이터로 분리하였다.

Table 2. Filtered Sample Dataset

| Sentence | Label# |
|---|---|
| Show me the step count for two weeks | 4 |
| Show me how much I walk for a month | 3 |
| Compare this week step count and last week | 6 |
| Tell me daily step count | 1 |
| Let me know this month's step count | 4 |
| Let me know weekly statistics for weekly steps | 2 |
| Step counts between July 9 and July 17 | 5 |
| Compare this week and last weeks step count by weekdays | 6 |
| Let me know weekly step counts | 2 |
| Let me know monthly step counts | 4 |
| Let me know step counts by weekdays | 1 |
| Compare last wek and this weeks step counts | 6 |
| Show me last week's step count | 5 |
| Compare last month and this month's step counts by day | 6 |
| Show me average monthly step counts 월별 | 3 |
| Show me this month's step conuts | 4 |
| Show me March 5 step counts by graph 3월 | 6 |
| How much did I walk last month? | 5 |
| Show me step counrs for the past 5 days | 4 |
| July's step counts | 5 |

라벨링 작업을 거친 후, 오픈 소스 한국어 처리기(Open Korean Text)를 통해 데이터 토큰화 작업을 진행하였다. OKT는 스칼라로 쓰인 한국어 처리 라이브러리로, OKT를 통해 데이터셋에 있는 문장들을 토큰화하여 총 838개의 토큰으로 구성된 토큰 파일을 생성하였다. 이후 토큰 파일을 기준으로 훈련 데이터와 테스트 데이터를 시퀀스 타입으로 변환하고 벡터화를 진행하여 모델에 입력값으로 사용하도록 하였다.

2) 분류 모델 설계

분류 모델은 Fig. 5와 같은 형태로, 총 4개의 Layer로 구성되어 있으며, 입력층과 중간층에서 사용한 활성화 함수 ReLu(Rectified Linear Unit)를 통해 연산이 이루어진다.

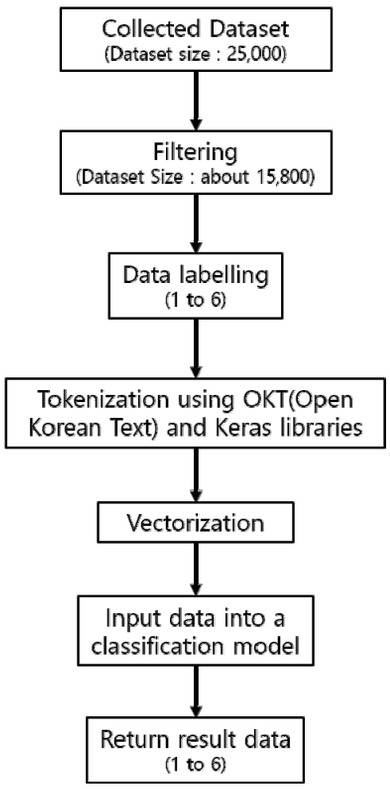



Fig. 4. Classification Model Process

그리고 마지막 출력층에서는 여러 카테고리 값을 예측할 때 주로 사용하는 Sotfmax 함수를 통해 1~6의 값 중 확률적으로 가장 높은 값을 출력한다.

| Layer (type) | Output Shape | Param # |
|---|---|---|
| dense_125 (Dense) | (None, 512) | 20992 |
| dense_126 (Dense) | (None, 256) | 131328 |
| dense_127 (Dense) | (None, 64) | 16448 |
| dense_128 (Dense) | (None, 8) | 520 |

Total params: 169,288
Trainable params: 169,288
Non-trainable params: 0

Fig. 5. Classification Model Summery

3) 모델 훈련 및 결과

모델 컴파일 시 사용된 옵티마이저는 Adam(Adaptive momentum estimation)으로, Equation (1)과 같다. RMSProp과 Momentum 방식을 합친 알고리즘으로, Momentum 방식과 유사하게 지금까지 계산해온 기울기의 지수 평균을 저장하며, RMSProp과 유사하게 기울기의 제곱값의 지수 평균을 저장한다[17-19]. $\beta_1$은 모멘텀 감소 비율, $\beta_2$는 Adaptive term 감소 비율을 말하며, n≥0 인 정수에 대해서 모멘텀 term $a_{n+1}$과 Adaptive term $b_{n+1}$을 정의한다. $a_{n+1}$은 지수이동평균을 사용하기 때문에 $\nabla f$의 계수$(1-\beta_1)$가 작은 값을 갖는다. 따라서 Adam Optimizer는 초기 update 속도를 보정(올리기)하기 위해서 상수 $\hat{a}_{n+1}$, n≥0를 다음과 같이 정의하여 변수를 update하는데 사용한다. 모멘텀 term $a_n$과 adaptive term $b_n$은 모두 벡터를 나타낸다. n〉〉1 인 경우 $\hat{a}_n$은 $\alpha$로 수렴한다. 마지막으로 변수 $x_{n+1}$을 정의한다.

$$
\begin{aligned}
a_{n+1} &:= \beta_1 \cdot a_n + (1-\beta_1)\cdot \nabla f(x_n), \quad a_0 := 0 \\
b_{n+1} &:= \beta_2 \cdot b_n + (1-\beta_2)\cdot \nabla f(x_n) \odot \nabla f(x_n), \quad b_0 := 0 \\
\widehat{\alpha}_{n+1} &:= \alpha \cdot \frac{\sqrt{1-(\beta_2)^{n+1}}}{1-(\beta_1)^{n+1}} \\
x_{n+1} &:= x_n - \frac{\widehat{\alpha}_{n+1}}{\sqrt{b_{n+1}}+\varepsilon} \odot a_{n+1}
\end{aligned} \tag{1}
$$

손실 함수는 데이터를 모델에서 6개의 카테고리로 분류하기 위해 Sparse Categorical Crossentropy를 사용하였으며, 이는 범주형 변수를 예측할 때 주로 사용한다. 모델 학습에서 사용된 Batch size는 256, 검증 데이터는 훈련 데이터에서의 20%로 지정하여 500 에포크로 학습을 진행하였다.

이후 테스트 데이터셋을 통한 정확도 측정 결과 약 0.89의 정확도를 보였다. Fig. 6을 참고하면 80~90번째 에포크까지는 train, test 데이터셋 모두 계속해서 loss가 감소하며, 이후부터 train 데이터셋에서는 지속적으로 감소하지만, test 데이터 셋에 대해서 loss가 다시 증가하는 양상을 보였다.

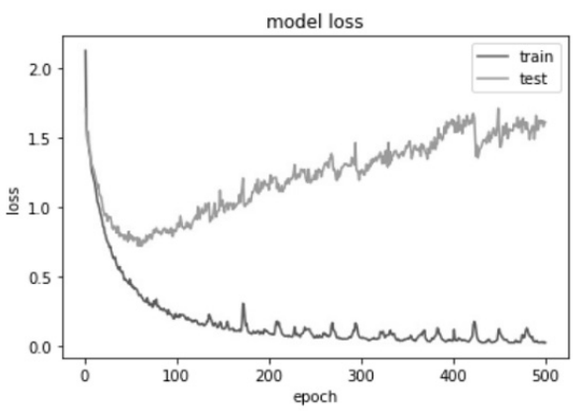



Fig. 6. Graph of Training and Validation Losses

### 3.4 2차 쿼리 생성 모델

1) 데이터 전처리

Fig. 7은 분류 모델의 전체적인 프로세스를 나타낸다. 1차 분류 모델에서 사용되었던 필터링된 데이터셋에 시간과 관련한 데이터셋을 임의로 추가하여 쿼리 생성에 필요한 월, 일에 대한 예측을 모델에서 잘 학습할 수 있도록 튜닝 작업을 진행하였다. 그리고 이렇게 만들어진 데이터셋의 문장들에 대해 Target 문장들(WHERE 절)을 생성해 데이터셋을 구축하였다.

챗봇 기반의 데이터 시각화 인터랙션을 위해 쿼리 생성에 초점을 두고 모델을 설계 및 개발하였기 때문에, 결과 데이터

는 복수 개의 데이터를 출력해야 하는 쿼리를 생성하므로 SELECT * FROM Table까지는 부분적으로 고정으로 하며, WHERE 절에 대한 생성만을 고려하고 있다. WHERE 절은 SQLite 문법 중 'BETWEEN' 구문을 통해 시간을 계산하도록 하였으며, date 함수를 통해 각 질문에 대한 WHERE 절 생성 및 라벨링 작업을 진행하여 Sentence-WHERE 페어 형태의 데이터 셋을 구축하였다.

이렇게 구축된 데이터셋에서 타겟이 되는 각 WHERE 절 데이터의 앞에는 〈SOS〉 토큰으로 탭(\t), 제일 끝에는 〈EOS〉 토큰으로 줄 내림(\n) 토큰을 추가하는 전처리 과정을 거치도록 하였다. 이후 분류 모델과 유사한 과정으로 토큰화 작업을 진행하였다. 총 생성된 토큰 1,040개 중 1,000개만 사용하였으며, 데이터를 시퀀스 데이터로 변환한 후 벡터화 과정을 거쳐 모델의 입력값으로 사용할 수 있도록 하였다.

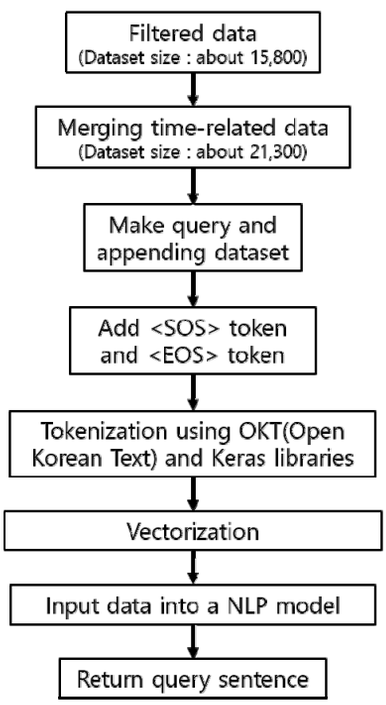



Fig. 7. Generation SQL Model Process

2) 데이터베이스 및 쿼리 생성 모델 설계

본 연구에서는 쿼리 실행이 가능하도록 SQLite 기반으로 데이터베이스를 구축하였다. 테이블은 'chat_stepcount_data'로 명명하고, 테이블에서의 컬럼(Column)은 걸음 수가 측정된 시간을 표기하기 위한 일일 단위 컬럼의 saved_time과 걸음 수를 저장하기 위한 stepCount 총 2개의 컬럼을 가지며, 샘플 데이터는 3월 1일부터 8월 20일까지의 임의의 걸음 수 데이터를 생성하여 테이블을 구축하였다(Fig. 8 참조).

쿼리 생성 모델에서는 신경망 기계 번역(Neural Machine Translation)에 주로 사용되는 Attention 모델을 통해 쿼리의 WHERE 절을 생성하고자 하였다. RNN(Recurrent Neural Network)에 기반한 Seq2Seq 모델은 인코더에서 입력 시퀀스에 대해 하나의 컨텍스트 벡터로 압축하여 표현하고, 이를 통해 디코더에서 출력 시퀀스를 만들어 내는 방식이었기 때문에 정보 손실이 발생하고 기울기 소실(Vanishing gradient) 문제가 발생하였다.

```
sqlite> select * from chat_stepcount_data;
6312|2020-03-01 00:00:00.000
9152|2020-03-02 00:00:00.000
5848|2020-03-03 00:00:00.000
4786|2020-03-04 00:00:00.000
6981|2020-03-05 00:00:00.000
7153|2020-03-06 00:00:00.000
9215|2020-03-07 00:00:00.000
11743|2020-03-08 00:00:00.000
8146|2020-03-09 00:00:00.000
8423|2020-03-10 00:00:00.000
6895|2020-03-11 00:00:00.000
7152|2020-03-12 00:00:00.000
8941|2020-03-13 00:00:00.000
9512|2020-03-14 00:00:00.000
11095|2020-03-15 00:00:00.000
6899|2020-03-16 00:00:00.000
8460|2020-03-17 00:00:00.000
8016|2020-03-18 00:00:00.000
7356|2020-03-19 00:00:00.000
8561|2020-03-20 00:00:00.000
8469|2020-03-21 00:00:00.000
7915|2020-03-22 00:00:00.000
13815|2020-03-23 00:00:00.000
7488|2020-03-24 00:00:00.000
9414|2020-03-25 00:00:00.000
6905|2020-03-26 00:00:00.000
10629|2020-03-27 00:00:00.000
6992|2020-03-28 00:00:00.000
4896|2020-03-29 00:00:00.000
9457|2020-03-30 00:00:00.000
8768|2020-03-31 00:00:00.000
```

Fig. 8. Sample of Step Count Dataset

LSTM(Long-Short Term Memory)는 Hidden state에 cell-state를 추가하여 정보 손실과 기울기 소실 문제를 극복하기 위해 고안되었으며, 본 연구에서는 Seq2Seq 모델의 Encoder에서 LSTM 모델을 사용하여 입력 값을 처리하도록 하였다. Attention은 디코더에서 출력 단어를 예측하는 매 시점에서 인코더의 입력 문장 전체를 한 번 더 참조하여, 해당 시점에서 예측해야 할 단어와 관련 있는 입력 단어 부분에 조금 더 집중하게 되는 매커니즘을 가지고 있다. 따라서 Attention 매커니즘을 Decoder에 적용시켜 Seq2Seq 모델을 개발하였다.

3) 모델 훈련 및 결과

모델 컴파일 시에 사용했던 옵티마이저와 손실함수는 Keras 라이브러리에 있는 함수를 활용했으며, 옵티마이저는 분류 모델과 동일한 Adam을 사용하였다.

손실 함수는 Sparse Categorical Cross entropy를 사용하여 학습을 진행하였으며, 100 에포크를 학습하도록 설정하였다. 결과적으로 손실 값은 약 0.0050, 정확도는 99.884를 보였다. 테스트 데이터셋으로 확인해본 결과는 다음 Fig. 9와 같다. q는 데이터셋에서의 Sentence들로, 인코더의 입력 값이며, a는 Target으로, WHERE 절을 의미하고 p는 모델이 생성한 예측된 WHERE 절을 의미한다.

```
_
q:  ['7월 9일 부터 17일 까지 걸음 수 보여줘 Wn']
a:  ["Wt saved _ time between date (' 2020-07 - 09 ', ' localtime ') and date (' 2020-07 - 18 ', ' localtime ') Wn"]
p:  ["saved _ time between date (' 2020-07 - 09 ', ' localtime ') and date (' 2020-07 - 18 ', ' localtime ') Wn"]
_
q:  ['7월 걸음 수 일자 별로 알려줘 Wn']
a:  ["Wt saved _ time between date (' 2020-07 - 01 ', ' localtime ') and date (' 2020-07 - 01 ', '+ 1 month ', ' localtime ') Wn"]
p:  ["saved _ time between date (' 2020-07 - 01 ', ' localtime ') and date (' 2020-07 - 01 ', '+ 1 month ', ' localtime ') Wn"]
_
q:  ['최근 2 주간 걸음 수 보여줘 Wn']
a:  ["Wt saved _ time between date (' now ', '- 14 days ', ' localtime ') and date (' now ', ' localtime ') Wn"]
p:  ["saved _ time between date (' now ', '- 14 days ', ' localtime ') and date (' now ', ' localtime ') Wn"]
_
q:  ['지난 주 걸음 수 보여줘 Wn']
a:  ["Wt saved _ time between date (' now ', '- 14 days ', ' weekday 0 ',' localtime ') and date (' now ', '- 7 days ', ' weekday 0 ',
' localtime ') Wn"]
p:  ["saved _ time between date (' now ', '- 14 days ', ' weekday 0 ',' localtime ') and date (' now ', '- 7 days ', ' weekday 0 ', '
localtime ') Wn"]
_
q:  ['지지난 주 대비 지난주 걸음 수 보여줘 Wn']
a:  ["time between date (' now ', ' start of month ', ' localtime ') and date (' now ', ' localtime ')) or ( saved _ time between date
(' now ', '- 1 month ', ' start of month ', ' localtime ') and date (' now ', '- 1 month ', ' start of month ','+ 1 month ', ' localti
me ')) Wn"]
p:  ["saved _ time between date (' now ', '- 14 days ', ' localtime ') and date (' now ', ' localtime ')) or ( saved _ time between da
te (' now ', '- 7 days ', ' weekday 0 ', ' localtime ') and date (' now ', ' localtime ')) Wn"]
_
q:  ['지난 달 과 이번 달 걸음 수 비교 해줄래 Wn']
a:  ["( saved _ time between date (' now ', '- 14 days ', ' weekday 0 ',' localtime ') and date (' now ', '- 7 days ', ' weekday 0 ',
' localtime ')) or ( saved _ time between date (' now ', '- 7 days ', ' weekday 0 ', ' localtime ') and date (' now ', ' localtime '))
Wn"]
p:  ["( saved _ time between date (' now ', '- 14 days ', ' weekday 0 ',' localtime ') and date (' now ', '- 7 days ', ' weekday 0 ',
' localtime ')) or ( saved _ time between date (' now ', '- 7 days ', ' weekday 0 ', ' localtime ') and date (' now ', ' localtime
'))"]
_
q:  ['최근 3일 동안 걸은 걸음 수 그래프 보여줘 Wn']
a:  ["Wt saved _ time between date (' now ', '- 3 days ', ' localtime ') and date (' now ', ' localtime ') Wn"]
p:  ["saved _ time between date (' now ', '- 3 days ', ' localtime ') and date (' now ', ' localtime ') Wn"]
```

Fig. 9. Sample of Predicted Results of NLP Model

### 3.5 전체 프로세스 동작 시나리오

시스템의 전반적인 프로세스는 Fig. 10과 같다. 챗봇 환경에 있는 사용자에게 걸음 수 데이터를 통한 시각화를 보여주며, 챗봇 환경에서 사용자는 원하는 기간에 대한 걸음 수 데이터를 조회하기를 원하면 챗봇에 질문한다. 해당 질문은 1차 분류 모델을 통해 1~6 중에 하나의 값으로 출력된다. 1~3 이내의 값이면 사전에 정의되어있는 쿼리를 실행하고, 4~6 이면 사용자의 질의를 쿼리 생성 모델로 전달하여 쿼리를 생성하도록 하는 프로세스를 거친다.

이 프로세스를 기반으로 한 시나리오는 다음 Fig. 11을 예로 들어 설명할 수 있다. 사용자가 일별 그래프를 보고 있는 상황에서 특정 기간에 대해 질문을 하는 경우, (a)는 6월의 걸음 수 정보를 요청하고 있는 그림이며, 챗봇은 이를 시각화로 지원하기 위해 사용자 질의를 시스템 내부의 모델로 전달하여 분류 모델과 쿼리 생성 모델을 거쳐 쿼리를 생성하게 된다. (b)를 예로 들어, 분류 모델에서는 6의 값이 출력되어 쿼리 생성 모델을 통해 쿼리를 생성하게 된다.

Fig. 12를 참조하여 6월 그래프 출력을 위해 데이터베이스로부터 데이터를 불러오기 위한 쿼리 생성 흐름을 설명할 수 있다. "SELECT * FROM chat_stepcount_data WHERE saved_time BETWEEN date('2020-06-01', 'localtime') AND date('2020-06-01', '+1 month', 'localtime')"이며,

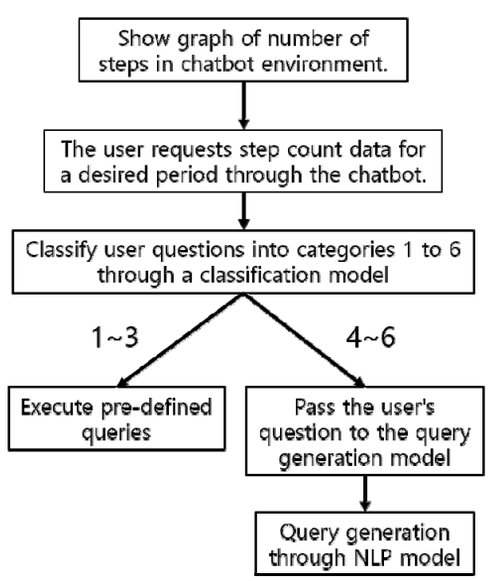
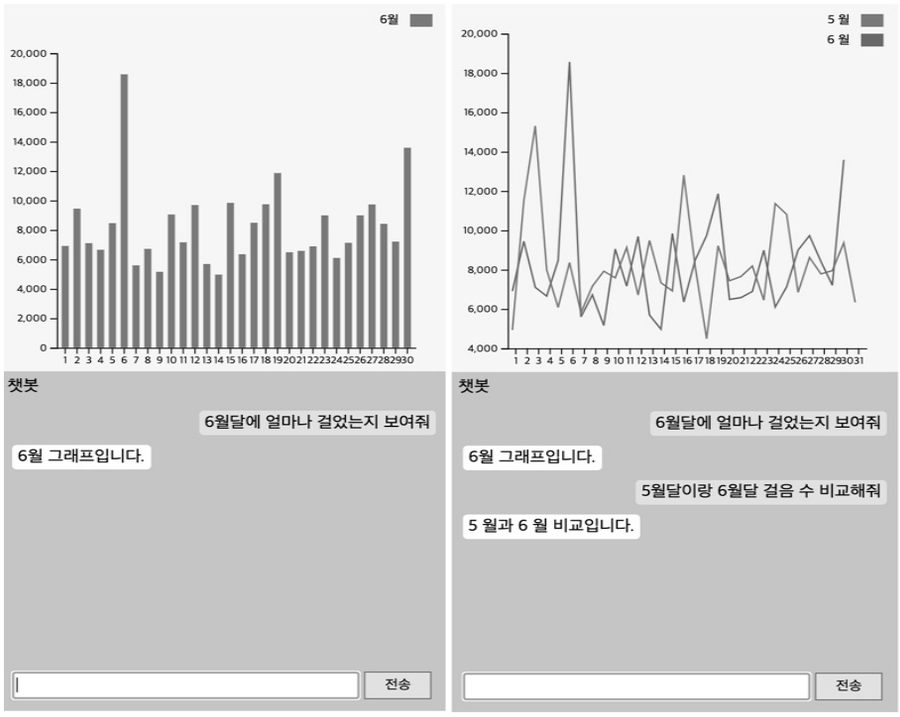



Fig. 10. System Process Flowchart

모델에서는 WHERE 절에 대한 부분만을 생성하여 SELECT * FROM chat_stepcount_data WHERE 부분과 연결하여 사용하도록 한다. (b)는 분류 모델에서 리턴 값으로 6을 받게 될 것이며, 쿼리 생성 모델을 통해 WHERE절의 (saved_time




(a) (b)

Fig. 11. Sample Screenshots Shown on a Mobile Device with Example Scenarios. (a) User Query to Show Steps for June and the Result in Bar Graph. (b) User Query to Compare Steps for May and June. The Result is Shown in a Line Graph.

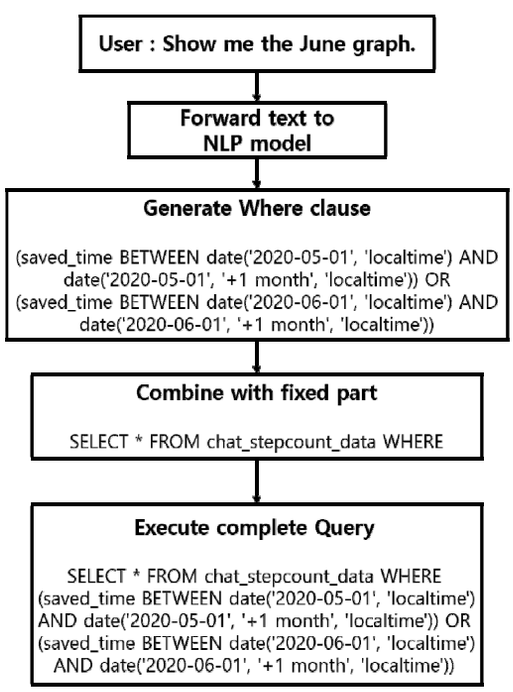



Fig. 12. Example of a query generation flow to output a June graph

BETWEEN date('2020-05-01', 'localtime') AND date ('2020-05-01', '+1 month', 'localtime')) OR (saved_time BETWEEN date('2020-06-01', 'localtime') AND date ('2020-06-01', '+1 month', 'localtime'))를 생성해낼 것이다. 이렇게 생성된 쿼리를 통해 출력되는 결과 데이터를 시각화로 그려 챗봇 환경에 있는 사용자에게 효과적으로 정보를 제공할 수 있다.

## 4. 결론 및 향후 연구

데이터가 급격히 증가함에 따라 헬스케어와 관련된 서비스들에서 개인화된 다양한 데이터를 활용하고자 한다. 이를 효과적으로 활용하여 사람들에게 보여주기 위해 데이터 시각화 기법이 활용되고 있으며, 시각화에서의 인터랙션 또한 강조되고 있다. 모바일 환경에서의 인터랙션은 제한된 화면의 크기로 인터랙션이 가능함을 인지하기 어려워 버튼 터치 방식이 주를 이루고 있으며, 앱에서 제공하는 제한된 내용의 시각화만을 볼 수 있다. 모바일 환경에서의 인터랙션 한계를 극복하기 위해, 챗봇과의 대화를 통해 데이터 시각화 인터랙션을 가능하게 하여, 사용자들에게 개개인의 데이터를 다양한 시각화를 통해 확인할 수 있도록 돕고자 한다. 이를 위해서는 사용자의 질의를 쿼리로 변환하고, 주기적으로 데이터를 축적하고 있는 데이터베이스에서 변환된 쿼리를 통해 결과 데이터를 불러올 수 있어야 한다. 하지만 기존의 연구들에 비해 시각화를 기반으로 하여 사용자의 질의를 쿼리로 변환하는 연구에 대해서는 아직 이루어지지 않고 있는 상황이다.

본 논문에서는 사전에 데이터 시각화 기법이 정해진 상황에서의 쿼리 생성에 초점을 맞추고자 하였다. 인터랙션 태스크에 따라 필터링은 바 그래프, 비교는 라인 그래프를 지원하며, 지원하는 인터랙션 태스크는 x축에 대한 필터링, 두 개 데이터 집단에 대한 비교 태스크가 가능하도록 모델 개발을 진행하였다. 시각화를 통해 요청한 정보를 제공받을 수 있는 자연어 처리 모델을 개발하기 위해 1,000명을 대상으로 한 설문조사를 통해 약 15,800개의 학습 데이터를 수집하였으며, 분류 모델에서는 약 89%, 쿼리 생성 모델에서는 약 99% 정확도를 보였다.

분류 모델에서의 정확도는 만족스러운 결과를 보였지만, 자연어 처리에서 높은 정확도를 보였던 것은 제한적인 데이터셋으로 구축하였기 때문에 학습 시, 데이터 편향과 과적합이 아닐까 추측하고 있다. 모든 날짜, 기간에 대한 데이터셋을 포괄하고 있는 데이터셋이 아니었으며, 학습 데이터셋에 포함되어있지 않은 기간에 대한 쿼리 생성의 정확도는 확인해보지 못하였지만, 높은 정확도를 보이지는 않았다. 따라서 자연어 처리에서의 데이터셋은 다양한 데이터들을 많이 포함하고 있을 때, 좋은 학습 여건이 만들어질 것이고, 이는 자연스럽게 모델 성능의 향상으로 이어질 것이라 기대한다.

본 연구에서는 챗봇 환경에서의 시각화 인터랙션을 위한 모델로 제한하여 서술하였지만, 향후 실시간으로 다양한 시각화 업데이트를 통해 사용자에게 원하는 정보를 신속하게 제공하기 위한 플랫폼을 개발하고자 한다. 이를 위해 부족했던 기간 데이터셋을 보충하여 모델의 학습량과 정확도를 높이도록 할 것이며, Attention 매커니즘 뿐만이 아닌, 최근에 큰 이슈를 끌고 있는 BERT 모델의 적용을 고려해보고자 한다. 이번 연구에서는 플랫폼 개발을 위해 분류와 쿼리 생성 모델을 분리하여 개발하였지만, 쿼리 생성과 시각화의 업데이트 속도를 고려하여 하나의 모델로 추가적인 개발을 해볼 수 있을 것이다.

## References


[1] E. S. Kim, "Affecting factors of mobile health care service usage and efficient utilization plan," *Korean Journal of Health Education and Promotion*, Vol.34, No.2, pp.41-52, 2017.

[2] L. S. Lee, S. H. Lee, J. S. Jeong, and K. Y. Noh, "Psychological factors influencing continuous use of mobile healthcare applications," *Journal of Digital Convergence*, Vol.15, no.7, pp.445-456, 2017.

[3] J. H. Park, "Smart Health Care Prospective Market Trends and Strategies" [Internet], https://news.kotra.or.kr/user/reports/kotranews/20/usrReportsView.do?reportsIdx=10985

[4] Y. H. Park and J. Y. Yun, "Design guidelines for data visualization of smart band: Focused on fitbit," *The Korean Society of Science & Art*, Vol.30, pp.141-149, 2017.

[5] E. J. No, "Visualization study of healthcare data: Focusing on mobile healthcare services," M.A. Degree Thesis, University of Ewha Womans at Seoul, Korea, 2015.

[6] Liu, Zhicheng and J. Stasko, "Mental models, visual reasoning and interaction in information visualization: A top-down perspective," in *IEEE Transactions on Visualization and Computer Graphics*, Vol.16, No.6, pp.999-1008, 2010.

[7] K. Blumenstein, C. Niederer, M. Wagner, G. Schmiedl, A. Rind, and W. Aigner, "Evaluating information visualization on mobile devices: Gaps and challenges in the empirical evaluation design space," in *Proceedings of the Sixth Workshop on Beyond Time and Errors on Novel Evaluation Methods for Visualization*, 2016.

[8] J. Y. Kim, "Analysis and design of mobile chatbot interface," M.Eng Thesis, University of Seoul National at Seoul, Korea, 2017.

[9] B. J. Kim. "Cognitive and emotional UX design methodology for mobile chatbot," *Korea Science & Art Forum*, Vol.34, 2018.

[10] D. Y. Chang and C. K. Lee, "A study of use intention of chatbot using the extended theory of planned behavior: Focusing on the role of interaction," *Journal of Tourism and Leisure Research*, Vol.31, No.8, pp.433-454, 2019.

[11] V. Zhong, C. Xiong, and R. Socher, "Seq2SQL: Generating structured queries from natural language using reinforcement learning," CoRR, abs/1709.00103, 2017.

[12] Y. Sun, D. Tang, N. Duan, J. Ji, G. Cao, X. Feng, B. Qin, T. Liu, and M. Zhou, "Semantic parsing with syntax- and table-aware SQL generation", in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pp.361-372, 2018.

[13] P. Wang, T. Shi, and C. K. Reddy, "Text-to-SQL Generation for Question Answering on Electronic Medical Records," in *Proceedings of The Web Conference*, pp.350-361, 2020.

[14] I. Ko and H. Chang, "Interactive Visualization of Healthcare Data Using Tableau," *Healthcare Informatics Research*, Vol.23, No.4, pp.349-354, 2017.

[15] D. J. Janvrin, R. L. Raschke, and W. N. Dilla, "Making sense of complex data using interactive data visualization," *Journal of Accounting Education*, Vol.32, No.4, pp.31-48, 2014.

[16] J. S. Yi, Y. A. Kang, J. Stasko, and J. A. Jacko, "Toward a deeper understanding of the role of interaction in information visualization," *IEEE Transactions on Visualization and Computer Graphics*, Vol.13, No.6, pp.1224-1231, Nov.-Dec. 2007.

[17] DeepLearning.AI, Improving Deep Neural Networks: Hyperparameter tuning, Regularization and Optimization [Internet], https://ko.coursera.org/lecture/deep-neural-network/rmsprop-BhJlm
[18] D. P. Kingma, and J. Ba, "Adam: A Method for Stochastic Optimization," in arXiv e-prints, 2014.
[19] M. J. Kim, Adam Optimizer [Internet], http://mjgim.me/2018/01/22/adam.html

**오 상 헌**
https://orcid.org/0000-0003-3371-7270
e-mail : shoh9179@gmail.com
20019년 동의대학교 컴퓨터과학과(학사)
2019년 ~ 현 재 동의대학교 IT융합학과 석사과정
관심분야 : Natural Language Processing, Machine Learning & Deep Learning, Data analytics

**허 수 진**
https://orcid.org/0000-0001-6802-8390
e-mail : sjhur0417@gmail.com
2017년 ~ 현 재 동의대학교 산업ICT기술공학과 학사과정
관심분야 : Data analytics, Machine Learning, Deep Learning, AI (Artificial Intelligence), Chatbot

**김 성 희**
https://orcid.org/0000-0002-9716-8349
e-mail : sh.kim@deu.ac.kr
2006년 이화여자대학교 컴퓨터공학과(학사)
2008년 이화여자대학교 컴퓨터공학과(석사)
2014년 퍼듀대학교 산업공학과(박사)
2017년 ~ 현 재 동의대학교 산업ICT기술공학과 조교수
관심분야 : Data Visualization, Machine Learning