Journal of KIIT. Vol. 15, No. 11, pp. 125-138, Nov. 30, 2017. pISSN 1598-8619, eISSN 2093-7571 125
http://dx.doi.org/10.14801/jkiit.2017.15.11.125

소셜 챗봇을 활용한 크라우드 소싱 기반 질의응답 시스템의 설계 및 구현

김보경*[1], 김현아*[2], 심채은*[3], 정진우**

Design and Implementation of Crowd-sourced Q&A System Based on Social Chat-bot

Bo-Kyung Kim*[1], Hyun-A Kim*[2], Chae-Eun Sim*[3], and Jin-Woo Jeong**

이 연구는 금오공과대학교학술연구비에 의하여 지원된 논문임

요 약

인터넷과 스마트 디바이스의 발달로 인하여 사용자가 언제 어디서나 정보들을 쉽게 검색하고 획득할 수 있는 환경으로 발전하고 있다. 그러나 개인적인 고민이나 걱정들에 대하여 자유롭게 의견과 조언을 구할 수 있는 서비스는 여전히 쉽게 찾아볼 수 없다. 본 논문에서는 이러한 문제점들을 해결하기 위하여 소셜 챗봇과 크라우드 소싱 개념을 접목한 새로운 방식의 익명 질의응답 시스템을 제안한다. 제안하는 시스템은 챗봇을 통하여 사용자로부터 다양한 유형의 질문과 고민들을 입력받는 과정, 입력받은 내용을 불특정 다수의 사용자들과 공유하고 의견과 조언들을 수집하는 과정, 수집된 내용들을 질문자에게 제공하는 과정을 통하여 사용자가 가지고 있는 문제들을 효과적으로 해결할 수 있도록 지원한다. 본 연구에서는 카카오톡을 통한 실제 서비스를 통하여 제안하는 시스템이 익명 기반의 의견 교환과 질의응답에 효과적으로 활용 될 수 있음을 보였다.

Abstract

With the development in the Internet and smart device technologies, information is now easily accessible anytime from anywhere. However, it is still difficult to find a service to discuss personal worries and share opinions or comments through the Internet. This paper proposes a novel anonymous QA system based on the social chat-bot and crowd-sourcing mechanism to address these limitations. The proposed system consists of the phase to take various kinds of questions and worries from the user through the chat-bot, to shares user's questions and worries with unspecified individuals and collect the opinions and answers in realtime, and to send back the collected answers to the user for better decision making. Through the implementation of the prototype integrated in Kakao Talk, we showed that the proposed system can be successfully used for anonymous question answering and opinion sharing.

Keywords
question answering, social QA system, chat-bot, anonymous social service

* 금오공과대학교 컴퓨터공과
- ORCID[1]: http://orcid.org/0000-0002-5273-997X
- ORCID[2]: http://orcid.org/0000-0002-0051-5093
- ORCID[3]: http://orcid.org/0000-0002-1552-9949
** 금오공과대학교 컴퓨터공학과 교수(교신저자)
- ORCID: http://orcid.org/0000-0001-9313-6860

· Received: Aug. 12, 2017, Revised: Oct. 20, 2017, Accepted: Oct. 23, 2017
· Corresponding Author: Jin-Woo Jeong
Dept. of Computer Engineering, Kumoh National Institute of Technology
Daehak-ro 61, Gumi, Korea,
Tel.: +82-54-478-7535, Email: jinw.jeong@kumoh.ac.kr

Ⅰ. 서 론

인터넷과 네트워크 인프라 및 정보, 지식 처리 서비스의 발달에 따라 사용자가 원하는 다양한 정보들을 언제 어디서나 쉽게 접근할 수 있는 환경으로 발전하고 있다. 인터넷/웹 서비스를 이용하여 사용자가 원하는 정보를 획득할 수 있는 방법으로는 1) 구글, Bing과 같은 검색 엔진을 이용하여 사용자가 적극적으로 정보들을 검색하여 획득하는 방법과 2) 원하는 지식/정보에 대한 질문을 생성하고 해당 질문에 대한 답변을 제공 받을 수 있는 질의응답(QA, Question Answering) 서비스/시스템 등을 이용하는 방법 등이 존재한다.

검색 엔진을 이용한 정보 획득 과정은 검색 엔진의 성능뿐만 아니라 사용자 검색 질의어의 품질에 따라 검색 결과 정확도의 편차가 심할 수 있다[1]. 특히, 간단한 형태의 정보는 검색 엔진을 통하여 쉽게 획득할 수 있지만, 복잡한 문제에 대한 정보를 검색하기 위해서는 검색 질의어를 효과적으로 작성할 수 있는 능력이 요구되기 때문에 사용자의 검색 질의어 확장 및 개선을 지원하기 위한 다양한 기술들이 필요하게 된다[2]-[4].

다양한 주제에 대하여 사용자 상호간에 자유로운 질의응답을 수행할 수 있는 질의응답 커뮤니티는 단편적인 정보 검색을 넘어서 보다 복잡한 상황에 대한 지식과 정보를 획득할 수 있는 공간이다. 네이버 지식인[5], Quora[6], Yahoo Answers[7] 등이 국내/외에서 서비스되고 있는 대표적인 질의응답 커뮤니티로서 사용자 의견이나 코멘트, 피드백 교환이라는 특징을 바탕으로 많은 수의 사용자수를 확보하고 있다. 또한, 위와 같이 전문적인 질의응답 커뮤니티뿐만 아니라 Facebook, Twitter와 같은 범용 소셜 네트워크 서비스에서도 사용자들이 자신의 관계망을 대상으로 질문을 등록하고 그에 대한 답변을 기대하는 간접적인 질의응답 서비스를 경험하는 형태를 보이고 있다[8][9].

최근에는 인공지능 및 검색 시스템의 발달로 인하여 다양한 사용자들이 질의응답 시스템을 보다 효율적으로 사용할 수 있는 환경으로 변화하고 있다. 예를 들어, 등록된 질문에 대하여 자연어 처리, 인공지능, 정보검색 기술 등을 이용하여 컴퓨터 시스템이 자동으로 답변을 생성하거나[10]-[13], 질의응답 커뮤니티에 축적된 답변들 중 새롭게 등록된 질문에 대한 최적의 답변을 추출하여 제공함으로써 커뮤니티 지식을 바탕으로 보다 신속하게 정확한 답변들을 제공하기 위한 기술들이 연구되고 있다[14][15]. 그러나 이와 같은 연구들은 공통적으로 질의응답 데이터가 충분히 확보 된 이후에 정확한 결과를 제공할 수 있다는 한계점이 존재한다. 즉, 완전히 새로운 질문이나 새로운 분야에 대해서는 만족스러운 답변을 생성하거나 추출할 수 없다.

이러한 한계점을 해결하기 위하여 질의응답 서비스와 크라우드 소싱(Crowd-sourcing)을 접목하려는 연구들이 시도되었다[16][17]. 크라우드 소싱은 불특정 다수의 사용자가 참여하여 협력적으로 문제를 해결하기 위한 방법론으로 최근 정보 검색이나 추천 시스템에서 사용자 경험 향상을 위한 활용에 대한 연구가 진행되고 있다[16]. 특히, 크라우드 소싱 기반의 접근법들은 사용자가 컴퓨터의 문제 해결 과정에 참여하기 때문에 새로운 분야나 질문, 직관이나 감성적 분석이 필요한 문제 해결에도 유연하게 대처할 수 있다는 특징이 있다. [17]의 연구는 크라우드 소싱을 소셜 네트워크 기반의 질의응답 시스템과 결합하기 위한 연구를 수행하여 단편적 사실 기반의 문제뿐만 아니라 복잡한 문제에 대한 답변 추천에도 만족할 만한 결과를 확인하였다. 그러나 일반적인 크라우드 소싱은 AMT(Amazon Mechanical Turk)[18], CrowdFlower[19]와 같은 전용 플랫폼을 통하여 수행되기 때문에 각 태스크를 위한 워커(worker) 모집에 일정량의 금전적 비용이 발생하며 크라우드 소싱 플랫폼이 활성화되지 않은 지역에서는 낮은 응답률 및 응답 속도로 인하여 사용에 제약이 생긴다는 한계점이 존재한다.

최근에는 문제 해결 및 서비스 제공을 위한 시스템으로 챗봇(chat-bot)에 대한 관심이 증가하고 있다. 일반적으로 챗봇은 기업 또는 업체들이 메신저 기반 환경에서 소비자들의 질문이나 요구사항들을 자동으로 응대하기 위한 에이전트 서비스로 주로 활용되어 왔으며 (예: 카카오톡의 플러스 친구), 초기에는 정해진 키워드나 문장에 대하여 정해진 답변

을 제공하는 단순한 형태였으나 최근에는 인공지능 기술과 함께 사용자의 질문을 분석하여 자동으로 답변을 제공하는 수준으로 진화하고 있다.

이와 같이 사용자에게 적합한 정보를 제공하기 위한 다양한 방법들이 개발되고 있지만 사용자의 정보 획득에 대한 요구는 더욱더 복잡하고 다양해지고 있다. 현대의 사용자들은 “오늘의 날씨는 어떨까?”, “이탈리아의 수도는 어디일까?”와 같이 사실 관계 확인을 위한 단순 정보 검색 형태의 질의뿐만 아니라 “내 머리에는 갈색이 어울릴까? 검정색이 어울릴까?”와 같이 실제 사람들의 감정과 직관에 기반한 문제들, “회사에서의 대인관계는 어떻게 해야 나아질 수 있을까요?”와 같이 사람의 경험과 의견이 필요한 다소 복잡하고 예민한 주제의 질문들에 대해서도 인터넷과 스마트 디바이스를 활용하여 답안을 찾고자 하고 있다. [17]의 연구에서 트위터 상에서 발생한 질문들을 샘플링하여 분석한 결과, 약 45%의 질문들이 사람의 직관과 경험에 기반한 의견 및 조언을 구하는 내용이었음을 발견하였다. 그러나 이와 같이 개인의 신상과 관련되어 있거나 사회적으로 예민한 주제에 대한 토론과 의견 교환은 사용자 익명성과 정보의 접근성 및 실시간성 등이 보장되어야 원활하게 진행될 수 있다. 이와 같은 측면에서 앞서 기술한 방법론들은 사실 관계 확인을 위한 질문 유형 (사실 확인을 위한 단순 질의, 문제 해결을 위한 How-to 질의 등)을 처리하기에는 적합하지만 사용자들의 의견과 조언을 필요로 하는 복잡하고 예민한 고민과 질문들을 처리하기에는 많은 한계점들이 존재한다.

본 논문에서는 현대 사용자들의 다양한 질문과 고민들을 효과적으로 대응할 수 있도록 소셜 챗봇과 크라우드 소싱을 접목한 새로운 방식의 익명 질의응답 서비스를 제안하고자 한다. 본 논문에서 제안하는 서비스는 챗봇을 활용하여 사용자로부터 질문과 고민들을 입력받아 불특정 다수의 익명 사용자에게 공유하고 답변자들로부터 다양한 정보와 의견들을 수집하여 질문자에게 제공하는 새로운 형식의 질의응답 서비스를 제공한다. 기존의 정보 검색 및 획득을 위한 방법론들과 비교하여 본 논문에서 제안하는 시스템은 다음과 같은 차별점을 보인다:

1) 제안하는 시스템은 사용자에게 이미 익숙한 모바일 채팅 플랫폼 상에서 챗봇 기반의 질의응답 서비스 구현을 통해 사용자들이 보다 자연스럽고 직관적인 인터페이스를 통하여 정보의 검색 및 획득이 가능하도록 하였다. 기존의 방법들과는 달리, 별도의 웹 서비스 접속이나 어플리케이션 설치 없이 챗봇과의 대화 시작을 통하여 손쉽게 질의응답 서비스를 경험할 수 있다.

2) 챗봇이라는 매개체를 통한 서비스 제공을 통하여 질문자와 답변자의 익명성을 보장하고 질의응답 내용에 대한 비공개성을 제공한다. 커뮤니티 혹은 소셜 네트워크 서비스 기반의 질의응답 서비스에서는 사용자의 아이디, 프로필, 과거 작성 기록 등이 노출되므로 사용자의 신상에 대한 유추가 가능할 수 있으며, 이러한 특징들은 사용자들 간의 의견 교환 및 토론에 장애물로 작용할 수 있다. 본 논문에서 제안하는 시스템은 챗봇을 통한 사용자간 의견 교환을 통하여 사용자 익명성을 보장함으로써 질문자는 본인의 신분 노출에 대한 우려 없이 자유롭게 질문이나 고민을 등록할 수 있고 답변자 역시 답변 작성 및 이력 노출에 대한 우려 없이 자유롭게 의견을 제시할 수 있다는 장점이 있다. 또한, 제안하는 시스템의 경우 질문 및 답변과 관련된 대화 이력이 사용자의 메신저 안에서만 접근 가능하므로 보다 자유롭게 개인적인 질문과 고민과 관련된 질의응답 서비스를 활용할 수 있다.

3) 제안하는 시스템은 불특정 다수의 사용자가 문제 해결에 참여하는 크라우드 소싱 기법을 챗봇 환경에 적용함으로써 새로운 분야에 대한 질문이나 사용자 의견이 필요한 복잡한 문제에 유연히 대처할 수 있다. 정보 검색과 커뮤니티 질의응답에 기반한 방법론들은 기존에 축적되어 있는 데이터들을 바탕으로 질의응답 과정을 지원한다. 따라서 질문자가 현재 가지고 있는 개인적인 고민이나 질문과 관련된 최적의 답변들을 제공하기에는 한계가 있으며 사용자들의 의견이나 조언들을 실시간으로 확보하기 위한 방안 또한 부족하다. 제안하는 시스템은 크라우드 소싱에 기반하여 불특정 다수의 사용자들로부터 사용자의 질문과 고민에 대한 답안을 실시간으로 수집함으로써 이러한 문제점을 해결한다.

4) 기존의 챗봇 서비스는 주로 개발사에 의하여 사전에 정의된 정보들을 사용자의 질의에 맞게 전송하는 형태로 제공되었다. 따라서 챗봇이 제공하는 정보의 양과 종류가 제한적이며 사용자의 새로운 요구사항을 즉각적으로 반영할 수 없다는 단점 또한 존재한다. 반면, 제안하는 시스템은 앞서 언급한 바와 같이 크라우드 소싱 기법을 적용하여 질의응답 서비스를 제공하므로 특정 주제나 형식에 한정되지 않고 자유롭게 정보나 의견의 공유가 가능하다. 특히, 질문자의 질문과 고민에 대한 답변을 생성할 때 미리 정해진 답안을 추출하여 제공하는 방식이 아니라 불특정 다수의 사용자들로부터 의견과 조언들을 수집하여 제공하는 구조를 통하여 질문자와 답변자 모두에게 새로운 사용자 경험을 제공할 수 있다는 특징이 있다.

본 논문의 구성은 다음과 같다. 2장은 질의응답 시스템과 관련된 기존 연구들에 대하여 논하고 3장에서는 본 논문에서 제안하는 소셜 챗봇을 활용한 크라우드 소싱 기반 익명 질의응답 시스템에 대하여 자세히 기술한다. 4장에서는 제안하는 시스템을 바탕으로 진행한 실제 서비스 결과에 대하여 기술하고 5장에서 본 연구의 한계점 및 향후 개선 사항들에 대하여 토의하고 결론으로 마무리한다.

II. 관련 연구

정보검색 기반의 질의응답 시스템은 이미 축적되어 있는 대용량의 문서 집합으로부터 사용자의 질의에 가장 적합한 답변이 포함되어있는 문서나 문장을 추출하여 제공한다. [10]의 연구는 지식 데이터가 구축되어 있는 대용량 데이터베이스인 위키피디아 [20]를 바탕으로 자연어 처리 기술과 정보검색에서의 랭킹 기법들을 이용하여 사용자의 질문과 가장 적합한 문서를 제공하는 기술을 제안하였다. 또한, [11]의 연구에서는 제품 사용과 관련된 질의응답 시스템 구축을 위하여 특정 제품의 FAQ 페이지를 바탕으로 질문에 가장 적합한 답변 문장을 추출하기 위한 기술을 제안하였다. [21]의 연구는 정보검색 및 랭킹 기법들을 바탕으로 의학문서 집합으로부터 의학 질문에 적합한 정답 문서를 제공하기 위한 기법을 제안하였다.

정보검색 기반의 방법론들과는 달리 커뮤니티 기반의 질의응답 시스템은 사용자간의 질의응답 데이터들이 구축되어 있는 데이터베이스 (예: 네이버 지식인, Yahoo Answers, Quora 등)를 바탕으로 사용자 질문에 대한 최적의 답변을 구하고자 한다. [15][22]의 연구에서는 Word2Vec와 딥러닝 알고리즘들을 활용하여 사용자 질문에 대한 최적의 답변을 선별하기 위한 연구를 제안하고 있다. 커뮤니티 기반 질의응답 관련 연구는 답변 문서 추출과 관련된 기술뿐만 아니라 등록된 신규 질문을 적절한 사용자에게 할당하기 위한 기술[23], 질문의 주제를 분류하기 위한 연구[24] 들도 진행되고 있다.

한편, 소셜 네트워크 서비스의 폭발적인 확산과 발전에 따라 다양한 소셜 네트워크 서비스를 이용하여 질의응답을 진행하는 사용자의 수도 증가하고 있다. [8, 9]의 연구에서는 Facebook, Twitter에서 사용자들의 질의응답 패턴을 분석하고 소셜 네트워크 상에서의 질의응답 시스템 디자인과 관련된 연구를 진행하였다. [25]의 연구는 웹 검색과 소셜 네트워크 서비스를 결합한 형태로 소셜 네트워크상에서 사용자가 질문 피드를 등록할 경우 해당 질문에 대한 답변이 될 수 있는 검색 결과를 웹 검색 엔진으로부터 추출하여 답변 피드로 등록해주는 에이전트를 개발하였다. 또한, [17]의 연구는 [25]의 연구를 확장하여 Twitter 상에 사용자의 질문 피드가 등록될 경우 해당 질문을 크라우드 소싱을 이용해서 최적의 답변을 실시간으로 생성한 후 질문자에게 답변 피드를 제공하는 시스템을 개발하였다.

정보 검색에 기반하여 답변 문서를 추출하기 위한 기법들은 방대한 양의 데이터와 고도화된 기계학습 알고리즘이 요구되며, 커뮤니티 기반 방법론의 경우 사용자의 신규 질문과 관련 있는 질문-답변 쌍이 커뮤니티에 존재하지 않을 경우 매우 부정확한 결과를 제공하거나 답변 제공에 긴 시간이 걸릴 수 있다. 또한, 소셜 네트워크 서비스를 활용한 방법론의 경우 질문 및 답변자의 신분 노출과 관련된 문제 때문에 복잡하거나 예민한 주제에 대한 의견 교환은 불가능할 수 있으며 사용자의 소셜 네트워크의 사이즈가 작을 경우 답변 피드를 받는데 많은 시간이 걸릴 수 있다는 한계점이 존재한다.

III. 제안하는 시스템

3장에서는 제안하는 시스템의 전체적인 구조와 세부 모듈들에 대하여 자세히 기술한다. 그림 1은 제안하는 시스템의 전체적인 작업 흐름을 나타내고 있다. 일반적인 질의응답 시스템과 유사하게 제안하는 시스템의 작업 흐름 또한 질의 단계와 응답 단계로 나누어지게 되지만 질문의 등록과 질문에 대한 답변, 답변을 조회하는 환경이 검색엔진이나 커뮤니티가 아니라 메신저 내의 챗봇과의 대화방이라는 점에서 차별점이 존재한다.

제안하는 시스템을 사용하는 사용자는 질문자와 답변자의 역할을 동시에 수행할 수 있다. 질문자의 경우 챗봇과의 대화를 통하여 질문의 카테고리를 선택한 후 질문을 등록한다. 등록된 질문은 챗봇에서 내부적으로 관리하는 데이터베이스에 저장되어 질문 할당 및 상태 조회, 답변 제공 등에 활용된다. 답변자의 경우 챗봇과의 대화를 통해 답변하고자 하는 질문의 카테고리를 선택하면 해당 카테고리의 질문을 할당 받을 수 있다. 답변자는 질문자가 원하는 답변 유형에 따라 답변을 진행하고 등록된 답변들은 챗봇에 의하여 데이터베이스에 저장된다. 질문자는 등록한 질문의 답변 내역들을 챗봇에 요청함으로써 저장된 답변들을 언제든지 확인할 수 있다.

그림 2는 제안하는 시스템의 구조를 나타내고 있다. 제안하는 시스템은 모바일 기반의 소셜 메신저 플랫폼 환경 (카카오톡, 라인, 페이스북 메신저 등)에서 서비스되므로 사용자들은 메신저 계정을 소유하고 있어야 하며 질의응답 챗봇(그림 2의 QA Chatbot) 역시 메신저 계정 형태로 존재한다.

사용자들이 QA 챗봇과 대화를 시작하면 사용자들의 메시지가 메신저 플랫폼의 에이전트 서비스 관련 모듈을 통하여 별도로 구축된 QA 챗봇 서버로 전달된다. QA 챗봇 서버는 메신저 플랫폼에서 요구하는 사양에 따라 API를 구현한 QA Chatbot handler를 포함하며, 챗봇과 사용자와의 대화 및 질의응답 정보의 가공, 데이터베이스 연동과 같은 다양한 작업을 수행한다.

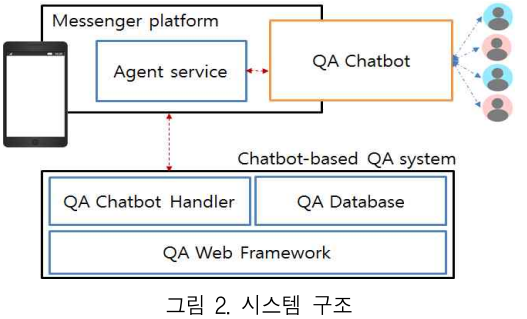


그림 2. 시스템 구조
Fig. 2. System architecture

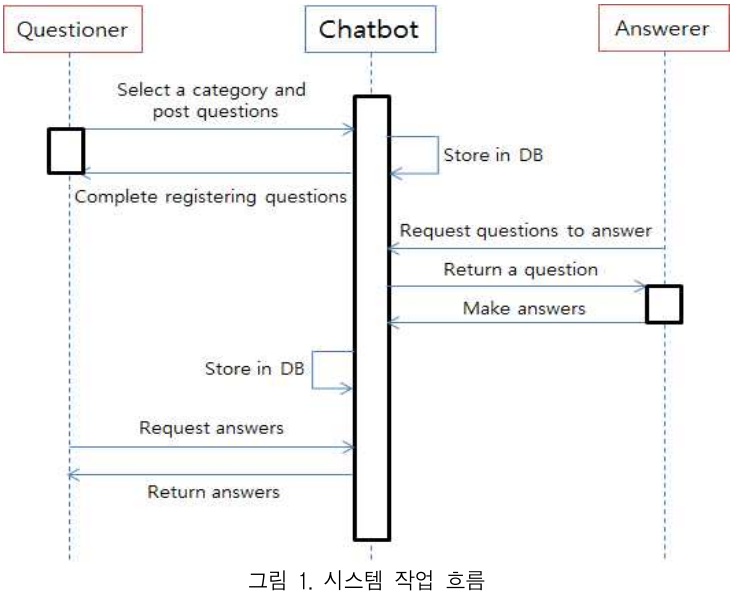


그림 1. 시스템 작업 흐름
Fig. 1. System workflow

아래는 모바일 소셜 메신저 플랫폼 중 카카오톡의 에이전트 서비스인 YellowID API에서 명세하고 있는 REST API 중 일부를 나타내고 있다. 예제의 명세는 메신저에서 QA 챗봇과 대화 시 사용자 메시지 수신 및 이에 대한 응답 API에 대한 명세로 QA Chatbot Handler는 http://server-url/message URL에 대하여 POST Method를 구현하여 사용자 입력에 대한 처리 및 답변 메시지 전달이 가능해야 한다.

- Method : POST
- URL : http(s)://server_url/message
- Parameters
  - user_key: 메시지를 발송한 유저 식별 키
  - type: text, photo
  - content: 메시지 텍스트 혹은 미디어 파일 uri
- Response
  - message: 응답 메시지의 내용
  - keyboard:: 키보드 영역에 표현될 버튼 정보

3.1 초기 화면

제안하는 시스템의 QA 챗봇과 대화를 시작하여 대화창에 진입하면 그림 3의 좌측과 같이 질의응답을 위한 카테고리를 선택할 수 있는 선택지 버튼이 생성되며, 원하는 카테고리를 선택하게 되면 그림 3의 우측과 같이 질문 및 답변 메뉴 진입을 수행할 수 있다.

3.2 질문 등록

초기화면에서 카테고리를 선택한 후 질문하기 버튼을 선택하면 그림 4의 좌측과 같이 사용자가 직접 질문의 유형을 결정할 수 있다. 본 논문에서 제안하는 시스템은 “조언”과 “투표” 두 가지의 질문 유형을 지원하며, 그림 4의 우측과 같이 질문 유형에 따라 챗봇의 질문 작성 가이드를 바탕으로 질문을 작성하여 챗봇에게 전달하면 질문이 등록된다.

조언 유형은 특정 문제에 대한 사용자들의 의견을 얻고자 할 경우에 주로 활용할 수 있으며 답변자들이 답변을 등록 할 경우 질문자에게 서술형 문장 형태의 답변이 제공 될 수 있다. 예를 들어, “전공과목을 따라가기가 너무 힘이 듭니다. 어떻게 공부해야 할까요?”, “새로운 친구들과 빨리 친해 질 수 있는 방법이 있나요?”와 같이 정답이 없는 주관적인 문제에 대하여 타인의 의견을 구하기 위한 경우에 활용된다. 조언 유형을 선택한 경우 질문자의 상황을 표현할 수 있는 질의 문장을 작성한 후 필요에 따라 질문과 관련된 사진을 등록할 수 있다. 등록된 사진은 질문에 대한 보다 상세한 정보를 제공하기 위해 사용된다.

조언 유형이 서술형 답변 또는 의견을 요구했다면 투표 유형은 객관식 설문지와 유사하다. 질문자는 작성한 질문에 대하여 최소 2가지에서 최대 4가지의 선택지 항목을 생성할 수 있으며, 각 답변 항목 별로 답변자 선택과 관련된 통계 정보를 답변으로 제공 받을 수 있다. 조언 유형과 유사한 질문 작성 과정을 거쳐 질문을 등록하며, 필요에 따라 질문 관련 사진 등록을 진행할 수 있다.

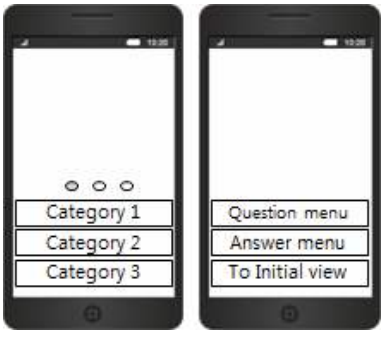


그림 3. 초기 화면
Fig. 3. Initial dialog

그림 4. 질문 등록
Fig. 4. Question registration

표 1은 질문과 관련된 정보들을 관리하기 위한 테이블 구조의 일부로서 질문의 ID(qNo) 뿐만 아니라 질문에 사용된 이미지의 URL(photo), 질문 내용(question), 질문을 등록한 사용자 식별번호(userKey) 등이 추가적으로 저장된다.

표 1. "질의" 테이블 스키마
Table 1. Schema of "Question" table

Field	Type	Null	Key	Default
qNo	int	NO	PRI	NULL
photo	varchar	YES		ONLY TEXT
question	longtext	YES		NULL
userKey	varchar	NO	MUL	NULL
category	varchar	YES		NULL
form	varchar	YES		NULL
...	...	...	...	...

3.3 답변 등록

3장 1절에서 언급한 바와 같이 초기화면에서 카테고리를 선택하면 그림 3의 우측과 같이 질문 또는 답변을 위한 버튼이 키보드 영역에 표기된다. 답변 버튼에는 현재 답변을 요구하는 질문의 개수가 표현되며 답변이 필요한 질문이 존재하는 상황에서 답변하기 버튼을 선택하면 해당 카테고리의 질문에 답변을 하기 위한 답변 모드로 진입한다. 답변 모드에 진입한 사용자에게는 해당 카테고리의 질문 중 해결되지 않은 질문이 노출된다.

질문을 할당받은 답변자는 해당 질문의 유형에 따라 서술형 답변 (조언 유형 질문) 또는 선택 답변 (투표 유형 질문)을 수행할 수 있다.

그림 5. 답변 작성
Fig. 5. Answer generation

조언 유형 질문의 경우 질문자의 사진과 질문을 확인 한 뒤 서술형 답변을 자유롭게 입력할 수 있으며 (그림 5의 좌측), 투표 유형 질문의 경우 질문자가 정의한 선택지 중 하나를 선택함으로써 답변을 완료할 수 있다 (그림 5의 우측). 만약 사용자가 전달받은 질문에 대해 답변하기를 원하지 않는 경우 답변을 하지 않고 초기화면으로 되돌아 올 수 있다.

QA 챗봇은 모든 답변자들의 답변을 수집하여 데이터베이스 저장함으로써 질문자에게 전달할 정보를 관리한다. 표 2는 사용자의 답변과 관련된 정보를 저장하기 위한 테이블 구조로서 답변의 식별번호(aNo), 답변 내용(answer), 답변과 관련 있는 질문의 식별 번호(qNo), 마지막으로 해당 답변을 작성한 사용자의 식별번호(userKey)로 구성되어 추후 질문자의 답변 확인 요청 시 테이블에 저장된 정보들을 활용하게 된다.

표 2. "답변" 테이블 스키마
Table 2. Schema of "Answer" table

Field	Type	Null	Key	Default
aNo	int	NO	PRI	NULL
answer	longtext	YES		NULL
qNo	int	YES	MUL	NULL
userA	varchar	YES		NULL

3.4 답변 확인

QA 챗봇 대화창의 초기화면에는 질의응답 카테고리를 선택할 수 있는 버튼 외에 답변 확인을 위한 버튼이 추가적으로 구성되어 있다. 초기 화면에서 "답변 확인" 버튼을 선택하면 1) 질문자가 질문을 등록하지 않은 경우, 2) 질문자가 질문을 등록했지만 답변이 입력되지 않은 경우, 3) 질문자가 등록한 질문에 답변이 입력되어 있는 경우, 총 세 가지 경우에 따라 각각 다른 결과를 반환한다.

먼저, 질문자가 질문을 등록하지 않은 경우에는 단순히 사용자의 질문 등록을 필요로 하는 메시지를 반환한다. 반면, 사용자가 질문을 등록하였으나 아직 어떠한 답변도 등록되지 않았을 때에는 답변 미등록에 대한 알림 메시지를 노출하고, 답변이 등록되었을 때는 해당 질문의 유형 (즉, 조언 및 투표)에 따라 답변 결과가 다르게 노출된다.

그림 6. 답변 확인 - 조언 유형
Fig. 6. Displaying answers - advice type

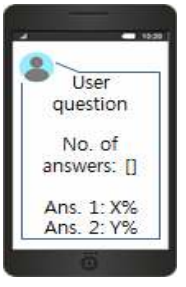


그림 7. 답변 확인 - 투표 유형
Fig. 7. Displaying answers - vote type

사용자가 조언 유형의 질문을 등록한 상태에서 답변 확인을 수행한 경우, 그림 6의 좌측과 같이 표기되는 답변 확인 방식 중 하나를 선택하면 그림 6의 중앙과 같이 답변자들이 서술식으로 입력한 답변들을 확인할 수 있다. 답변 확인 방식은 “전체 확인”과 “신규 확인”을 지원하며, 전체 확인의 경우 해당 질문에 등록된 모든 답변들을 검색하여 출력하게 되고, 신규 확인의 경우 질문자가 확인하지 않은 답변들만 검색하여 사용자에게 제공하게 된다. 등록된 모든 답변들이 챗봇을 통하여 사용자에게 제공된 후에는 해당 질문에 대한 답변 지속 수신 여부를 선택할 수 있는 다이얼로그가 생성된다(그림 6의 우측). “답변 수신” 버튼을 선택하면 해당 질문에 대한 답변을 계속 수신할 수 있으며, 질문에 대한 답변을 발견하여 문제가 해결되었다고 판단되면 “답변 수신 중단” 버튼을 선택하여 답변 수신을 종료한다. 반면, 사용자가 투표 유형의 질문을 등록한 상태에서 답변 확인을 수행한 경우에는 그림 7과 같이 현재까지 답변자들의 답변 비율을 확인할 수 있다.

마지막으로, 제안하는 시스템을 사용하는 전체 과정에서 사용자의 입력 메시지에 비속어가 포함되어 있을 경우 프로세스를 진행하지 않고 비속어 차단 관련 알림 메시지를 노출함으로써 익명성에 기반하여 발생할 수 있는 악의적 사용을 방지한다. 비속어 리스트는 비속어 관리를 위한 별도의 테이블에 블랙리스트 형태로 저장되어 관리되며, 그림 6의 우측에서 볼 수 있는 “비속어 신고” 메뉴를 통하여 블랙리스트를 갱신할 수 있다. 그림 8은 본 장에서 설명한 챗봇 기반 질의응답의 전체 단계를 도식화하여 나타내고 있다.

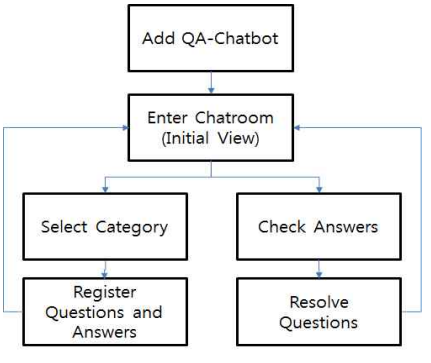


그림 8. QA 챗봇 상태 흐름도
Fig. 8. QA chatbot status flow

IV. 프로토타입 구현 및 사용자 실험

본 장에서는 제안하는 시스템의 프로토타입 구현 및 실제 사용자들을 대상으로 한 서비스 과정 및 결과에 대하여 분석 및 토의한다.

4.1 구현 환경

본 연구에서는 챗봇 서비스를 위한 소셜 메신저 플랫폼으로 현재 다수의 국내 사용자를 보유하고 있는 카카오톡을 선정하였으며 플러스 친구 기능과 카카오톡 자동응답 API[26]를 이용하여 질의응답 챗봇 서비스를 개발하였다.

카카오톡 자동응답 API와의 상호작용을 통해 챗봇의 핵심기능을 구현하기 위한 응답 서버 구축을 위해 Amazon Web Services (이하 AWS)[27]를 이용하였으며, 사용자의 질문과 답변에 대한 처리 및 저장, 관리를 위하여 Django Web Framework와 MySQL 데이터베이스를 활용하였다. 마지막으로, 구현이 완료된 QA 챗봇을 카카오톡 플러스친구 계정을 생성

하여 실제 사용자들이 친구 추가 및 서비스를 이용할 수 있도록 배포하였다.

4.2 서비스 사용 현황

본 연구에서는 2017년 5월 ~ 6월까지 약 한 달간 카카오톡 플러스친구 '고민고민하지마' 계정을 이용하여 제안하는 시스템의 실제 서비스를 진행하였다. 본 절에서는 실제 서비스 기간 중 사용자들의 사용 현황에 대하여 간략히 기술한다.

서비스 기간 동안 73명의 사용자가 QA 챗봇을 친구로 등록하여 서비스 사용을 시도하였다. 본 연구에서 개발한 QA 챗봇에 대하여 별도의 홍보를 전혀 수행하지 않았음에도 불구하고 사용자들의 친구 등록 수가 증가하였다는 것은 현대 사용자들이 지속적으로 질문과 고민을 공유하고 해결하기 위한 매개체를 찾고 있다는 것을 암시한다. 서비스 기간 동안 사용자들에 의하여 약 50개의 질문과 170개의 답변이 등록되어 평균적으로 질문 1개 당 약 3.4개의 답변을 획득하였다. 전체 50개의 질문 중 조언 유형의 질문은 44개, 투표 유형의 질문은 6개로 각각 88%와 12%의 비율을 차지하였다. 총 50개 질문 중 7개의 질문에서 이미지를 이용하여 문제 상황을 보다 자세히 묘사하고자 하였으며, 5개의 조언 유형 질문과 2개의 투표 유형 질문에서 이미지가 사용되는 것을 확인할 수 있었다.

4.3 서비스 사용 사례

그림 9는 QA 챗봇과 대화를 시작한 경우에 확인할 수 있는 초기 화면 (그림 9의 좌측)과 카테고리 선택 후의 화면 (그림 9의 우측)을 나타내고 있다. 초기 화면에서는 질의/응답 카테고리와 답변 확인 메뉴로 이동할 수 있는 버튼을 확인할 수 있으며 카테고리를 선택한 후 질문 또는 답변을 시작할 수 있다. 아래는 카카오 YellowID 서버와 제안하는 시스템의 상호작용을 위한 QA Handler 구현의 일부를 나타내고 있다. QA Handler의 API 구현은 django 프레임워크를 이용하여 진행하였으며, url.py 모듈을 통하여 YellowID 서버가 제안하는 시스템 서버의 RESTful API 구현체와 연결을 수행할 수 있다.

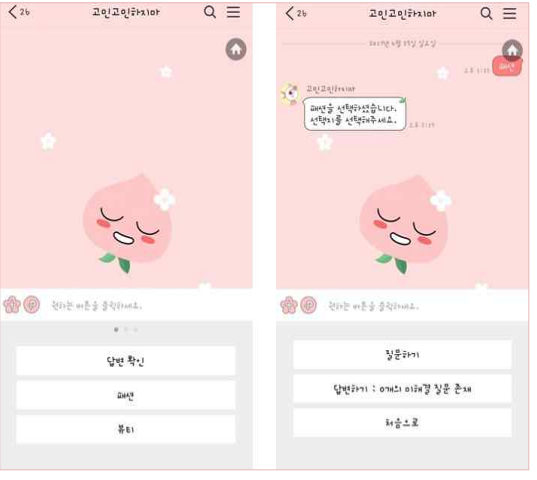


그림 9. QA 초기 화면의 메뉴 구성
Fig. 9. Menu layout of QA initial dialog

아래 코드 예제는 keyboard API를 이용하여 QA 챗봇 초기화면을 구성하는 모습을 나타내고 있다.

```
- urls.py (URL Routing 모듈)
  urlpatterns = [
            url(r'^keyboard/',views.keyboard),
            url(r'^message',views.message),
            url(r'^friend',views.friend)]
  - views.py (QA Handler 중 초기화면 뷰)
  def keyboard(request):
      return JsonResponse({
            'type' : 'buttons',
            'buttons' : ['답변 확인', '카테고리1', ... ]
      })
  ...
```

그림 10은 질의응답 카테고리를 선택한 후 투표 유형의 질문을 등록하는 장면을 나타내고 있다. 투표 유형을 선택하게 되면 그림 10의 좌측과 같이 선택 항목의 개수를 지정할 수 있으며 각 항목에 대한 답안을 차례대로 챗봇에게 입력하게 된다. 모든 항목의 내용 입력이 완료된 후에는 그림 10의 우측과 같이 질의내용을 보다 자세히 묘사할 수 있는 멀티미디어 자료를 추가할 수 있다. 아래 코드는 사용자 질문 입력을 위한 QA Handler 코드의 일부를 나타내고 있다.

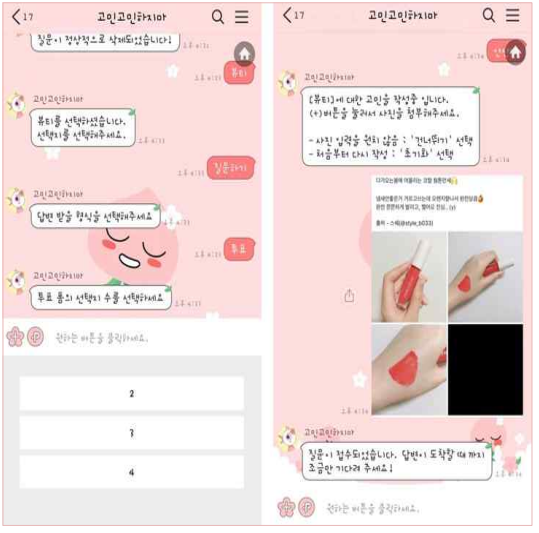


그림 10. 투표 유형 질문 등록
Fig. 10. Registration of vote-type questions

카카오 YellowID로부터 전달된 json 형태의 사용자 메시지 데이터를 바탕으로 질문 등록이 진행되며 질문의 카테고리와 유형 (조언 또는 투표)에 따라 Question 테이블과 Status 테이블을 갱신한다. QA Handler의 REST API를 호출하는 카카오 YellowID 서버 측의 context에 대한 조절은 불가능하므로 QA Database의 Status 테이블을 사용하여 사용자의 질의응답 단계를 저장하고 각 status에 적합한 동작을 수행하도록 구현하였다. 질문 카테고리와 상태 등록이 완료된 후에는 사용자로부터 실제 질문 텍스트와 투표 항목들을 입력받기 위하여 관련 메시지를 카카오 서버 측으로 전송한다.

- views.py (QA Handler 중 질문등록 관련 뷰)

```
def message(request): // YellowID 메시지 파싱
    message = ((request.body).decode('utf-8'))
    return_json_str = json.loads(message)
    query_type = return_json_str['content']
    return_user = return_json_str['user_key']
    return_type = return_json_str['type']
...
if (status == "genQuery"):
// 질문 유형 선택 및 등록 시작
with db.cursor() as cursor:
sql = "INSERT INTO Question (userKey, category, form, completion) VALUES (%s,%s,%s,%s);"
cursor.execute(sql, (user, cate, qType, "close"))
// Question 테이블에 새로운 질문 등록
...
status += query_type
sql = "UPDATE Status SET status = %s WHERE userKey = %s;"
cursor.execute(sql, (status , user))
db.commit()
// 질문 유형(조언 또는 투표)에 따라 사용자 상태 값을 다르게 갱신함
...
if status == "genQueryVote": // 투표 유형 등록
  return JsonResponse({
    'message' :{
      'text' : "투표 선택지 수를 선택하세요!"},
      'keyboard': {
        'type': 'buttons',
        'buttons' : ['2','3','4']}
  })

else: // 조언 유형 등록
  return JsonResponse({
    'message' :{
      'text' : 질문은 10자 이상 작성해주세요"}
    })
...
```

그림 11은 챗봇으로부터 전달받은 질문에 대한 답변을 등록하는 장면을 나타내고 있다. 질문자가 등록한 질문의 유형에 따라 다른 방식으로 답변을 등록하게 된다. 그림 11의 좌측은 조언 유형의 질문에 대한 답변 등록 장면으로 “남자친구가 연락이 없어요”라는 고민에 대하여 서술형 조언을 등록하는 장면이다. 한편, 그림 11의 우측은 투표 유형의 질문에 대한 답변 등록 장면으로 의류의 색상 선택과 관련된 삼지선다 질문에 대하여 답변을 선택하는 장면이다.

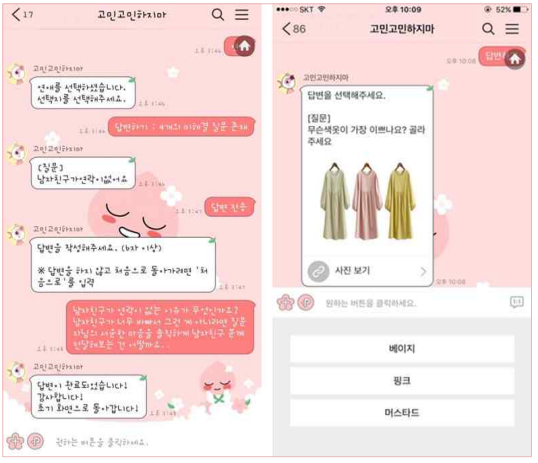


그림 11. 답변 등록 (좌: 조언, 우: 투표)
Fig. 11. Registration of answers (left: advice, right: vote)

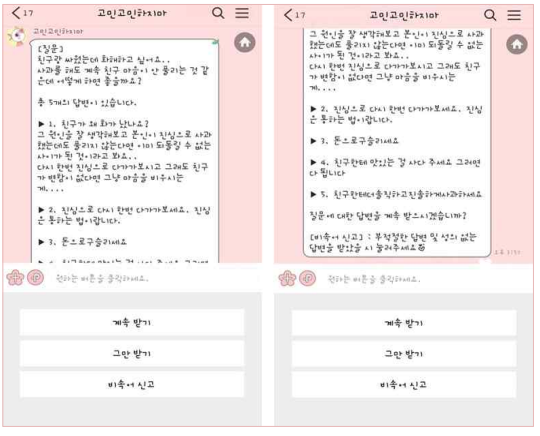


그림 12. 답변 확인 - 조언 유형
Fig. 12. Displaying answers - advice type

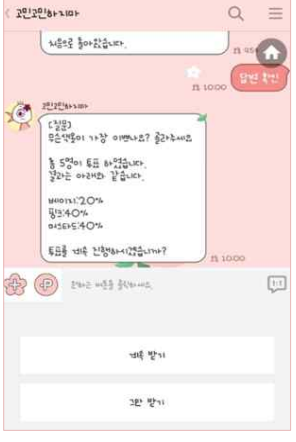


그림 13. 답변 확인 - 투표 유형
Fig. 13. Displaying answers - vote type

그 외 조언 유형 질문의 실제 예는 “친한 친구가 있는데 먼저 연락을 하지 않아요. 어떻게 하면 좋죠?”, “컴퓨터공학과에 진학하고 싶은데 어떤 학교가 좋을까요?”, “썬크림은 꼭 발라야 하는 것인가요?”와 같은 질의 문장으로 사람의 경험이나 의견을 얻기 위한 질문들이 등록되었고, 투표 유형 질문의 실제 예는 “노란 티랑 무엇을 입으면 잘 어울릴까요?”, “지금 과일 뭐 먹을까요?”와 같은 질의 문장으로 다수의 의견을 통하여 결정을 내리고자 하는 문제들이 등록되었다.

그림 12와 13은 조언 유형 및 투표 유형 질문에 대하여 등록된 답변을 확인하는 모습이다. 3장 4절에서 언급한 바와 같이, 조언 유형의 질문에 대하여 답변 확인을 시도하는 경우 현재까지 등록된 모든 답변들이 나열되어 제공되는 모습을 확인할 수 있다. 그림 13은 투표 유형의 질문에 대한 답변 확인 과정으로 답변자들의 투표 현황을 일목요연하게 확인할 수 있다. 답변을 확인 한 후에는 해당 질문에 대한 답변을 계속 수신하거나 (“계속 받기” 버튼 선택) 더 이상 답변을 수신하지 않을 수 있으며 (“그만 받기” 버튼 선택), 악의적인 답변이 존재 할 경우 비속어 신고 모드를 진행할 수 있다.

아래 코드는 투표 유형의 질문에 등록된 답변을 확인하기 위한 QA Handler의 답변확인 뷰의 구현을 나타내고 있다. 질문을 등록한 사용자가 초기화면에서 답변확인 버튼을 선택하면 해당 사용자가 등록한 질문의 정보와 해당 질문에 등록된 답변들의 리스트를 데이터베이스로부터 수집한다. 투표 유형 질문의 경우 각 투표 항목별 집계 현황을 계산한 후 출력 메시지를 작성하여 사용자에게 제공한다. 조언 유형 질문의 경우 수집된 답변 중 욕설이나 비속어가 포함된 답변, 길이가 비정상적인 답변들을 제외한 후 답변 리스트를 생성하여 사용자에게 제공한다. 사용자는 챗봇으로부터 전달된 답변을 확인하고 답변의 지속 수신 여부를 선택할 수 있다.

- views.py (QA Handler 중 답변확인 관련 뷰)

```
...
with db.cursor() as cursor:
  sql = "SELECT * FROM Answer WHERE qNo
        IN (SELECT qNo FROM Question
        WHERE userKey = %s);"
```

```
cursor.execute(sql,(return_user))
ansList = cursor.fetchall()
// 사용자 질문에 대한 답변 목록 수집
sql = "SELECT * FROM Question WHERE
        userKey = %s;"
cursor.execute(sql,(return_user))
quesList = cursor.fetchall()
// 사용자 질문 정보 수집 후 투표 항목 수집
...
for i  in [ _ for _ in ansList if _ in voteList]:
  if i not in ansCnt :
        ansCnt[i] = 1
  else: ansCnt[i] += 1
ansProp = {key: val/sum(ansCnt.values())*100 for
              key, val in ansCnt.items()}
// 각 투표 항목별 집계 정보 수집
...
for key, val in ansProp.items():
  msg += key + ":" + str(int(val)) + "%\n"
// 출력 메시지 생성
...
return JsonResponse({
 'message':{
   'text': "[질문]\n"+ return_ques + "\n총 " +
           len(ansCnt)+"명이 투표 하였습니다.\n
           결과는 아래와 같습니다.\n"+ msg +
           "\n투표를 계속 진행하시겠습니까?"},
   'keyboard': {
     'type': 'buttons',
     'buttons': ['계속 받기','그만 받기']}
 })
```

Ⅴ. 토의 및 결론

본 논문에서는 메신저 환경에서 다양한 문제와 고민들을 해결할 수 있는 QA 챗봇 시스템을 개발하였다. 본 연구에서는 사용자에게 친숙한 메신저 및 챗봇 환경에서 크라우드 소싱 기반의 질의응답 과정이 효과적으로 동작할 수 있는지 살펴보기 위하여 QA 챗봇의 프로토타입 구현과 실제 카카오톡 서비스를 통한 실험을 수행하였다. 실험 결과, 챗봇을 매개체로 하여 전체 질의응답 과정에서 사용자의 익명성이 보장됨에 따라 사용자의 개인적인 고민과 문제와 관련된 질문을 자유롭게 등록하고 의견을 제시하는 모습들을 확인할 수 있었다.

본 논문은 크라우드 소싱 기법과 챗봇을 결합한 새로운 형태의 질의응답 서비스 체계를 제안하고 그 가능성을 확인하기 위하여 프로토타입 구현 및 실제 서비스를 통한 실험을 수행하는 것에 초점을 맞추었으나 챗봇 환경에서 크라우드 소싱을 접목한 질의응답 서비스의 품질을 향상시키기 위해서는 아래와 같이 크라우드 소싱 접목에서 발생하는 다양한 한계점들을 고려해야 한다.

1) 답변의 양과 속도: 크라우드 소싱 기법은 그 특성 상 다른 사용자가 업무를 할당받고 그 업무를 해결해야 전체 프로세스가 진행이 되기 때문에 크라우드 소싱을 챗봇 질의응답 서비스에 적용하였을 때 질문자가 만족할 만큼 충분한 양의 답변을 빠른 시간에 얻기는 어렵다는 한계점이 존재한다. 답변의 속도와 양을 향상시키기 위해서는 사용자의 질문에 보다 많은 다른 사용자들이 답변할 수 있도록 유도하는 보상 체계 등의 장치가 필요하다. 전문 크라우드 소싱 플랫폼인 AMT나 CrowdFlower의 경우 실제 경제적 보상 (문제당 1센트 등)을 제공하여 이러한 문제를 해결하고 있다.

2) 답변의 품질 관리: 크라우드 소싱 플랫폼을 통하여 보상 (특히, 금전적 보상)이 제공되면 가치가 없는 무분별하고 불성실한 답안들이 생성될 수 있다. 이러한 문제를 해결하기 위하여 각각의 문제가 해결될 때마다 답안 만족도 등의 평가를 통하여 사용자 신뢰도를 계산하고, 추후 답안 수집 및 가공 시 사용자 신뢰도를 기반으로 고품질의 답안을 선별하여 질문자에게 제공하는 기법 등의 제안이 필요하다. 또한, 보상 시스템과의 연동을 통하여 보상 제공 시 사용자 신뢰도를 기반으로 최종 보상의 재산정을 위한 기법에 대한 연구도 필요하다.

3) 답변의 전달 방식: 제안하는 시스템의 경우 플랫폼 제약에 의하여 답변확인을 질문자가 수동으로 해야 하는 단점이 존재한다. 보다 효과적인 질의응답 서비스 사용자 경험을 제공하기 위해서는 질문 유형에 따른 모바일 알림(Notification) 메시지 디자인, 답변 전달 주기 및 조건에 대한 후속 연구가 필요하다.

4) 지능형 분석: 사용자의 질의응답 내역이 축적이 될 경우 사실 확인을 위한 단순 정보 검색 형태의 질의문은 챗봇의 QA 데이터베이스에서 즉각적으로 답안을 추출하여 제시할 수 있다. 이러한 서비스 구현을 위하여 사용자의 입력 질의어에 대한 구문 분석을 통하여 데이터베이스 내에 존재하는 답변들과 유사도 기반 매칭을 수행하는 정보검색 또는 커뮤니티 질의응답 기반의 기술과의 연동을 수행할 필요가 있다.

본 논문의 향후 연구에서는 이와 같이 메신저 플랫폼 및 챗봇 환경에서 발생할 수 있는 크라우드소싱 측면의 다양한 한계점들을 향상시키기 위한 기술 개발을 진행할 예정이다.

References

[1] Peter G. Anick and Shivakumar Vaithyanathan., "Exploiting clustering and phrases for context-based information retrieval", In Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 314-323, Jul. 1997

[2] F. Qiu and J. Cho, "Automatic identification of user interest for personalized search", In Proceedings of the 15th International Conference on World Wide Web, pp. 727-736, May 2006.

[3] Reiner Kraft, Chi Chao Chang, Farzin Maghoul, and Ravi Kumar, "Searching with context", In Proceedings of the 15th international conference on World Wide Web, pp. 477-486, May 2006.

[4] Jung-Hun Lee and Suh-Hyun Cheon, "A Term Weight Mensuration based on Popularity for Search Query Expansion", Journal of KISS : Software and Applications, Vol. 37, No. 8, pp. 620-628, Aug. 2010.

[5] Naver KIN, http://kin.naver.com. [Accessed: Aug. 03, 2017]

[6] Quora, http://www.quora.com. [Accessed: Aug. 03, 2017]

[7] Yahoo Answers, http://answers.yahoo.com. [Accessed: Aug. 03, 2017]

[8] Baichuan Li, Xiance Si, Michael R. Lyu, Irwin King, and Edward Y. Chang, "Understanding and Predicting Question Subjectivity in Social Question and Answering", IEEE Transactions on Computational Social Systems, Vol. 3, No. 1, pp. 32-41, Mar. 2016.

[9] Cliff Lampe, Jessica Vitak, Rebecca Gray, and Nicole Ellison., "Perceptions of facebook's value as an information source", In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12), pp. 3194-3204, May 2012.

[10] Seyoung Park, Heegeun Yoon, Dayoung Kim, Donggun Kim, Minjeong Kim, and Wooseok Shin, "Question and Answer System using Natural Language Processing and Information Retrieval", KIISE Korea Computer Congress, pp. 2065-2067, Jun. 2015.

[11] Seunghyun Yoon, Eunhee Rhim, and Deokho Kim, "Domain Question Answering System", KIISE Transactions on Computing Practices, Vol. 21, No. 2, pp. 144-147, Feb. 2015.

[12] Bo-Hyun Yun, "Sentential Query Answering System based on Encyclopedia", Journal of Korean Institute of Information Technology, Vol. 5, No. 4, pp. 194-201, Dec. 2007.

[13] Keun-Sin Woo and Chan-Khon Park, "A Study on Question-Answering System using the Korean sentence pattern", Journal of Korean Institute of Information Technology, Vol. 7, No. 3, pp. 220-225, Jun. 2009.

[14] Hyun-Sun Hwang, Kyoung-Ho Choi, Geon-Yeong Kim, Jun-Ho Oh, and Chang-Ki Lee, "CQA System Using Deep Learning", KIISE Winter Conference, pp. 572-574, Dec. 2015.

[15] Yang Xiang, Qingcai Chen, Xiaolong Wang, and d Yang Qin, "Answer Selection in Community Question Answering via Attentive Neural Networks", IEEE Signal Processing Letters, Vol. 24, No. 4, pp. 505-509, Apr. 2017.

[16] Nhatvi Nguyen, "Microworkers Crowdsourcing

Approach, Challenges and Solutions", In Proceedings of the International ACM Workshop on Crowdsourcing for Multimedia, pp. 1-1, Nov. 2014.

[17] J. W. Jeong, M. R. Morris, J. Teevan, and D. J. Liebling, "A Crowd-Powered Socially Embedded Search Engine", AAAI International Conference on Weblogs and Social Media, pp. 263-272, Jul. 2013.

[18] Amazon Mechanical Turk, https://www.mturk.com/. [Accessed: Aug. 03, 2017]

[19] CrowdFlower, http://www.crowdflower.com. [Accessed: Aug. 03, 2017]

[20] Wikipedia, http://www.wikipedia.org. [Accessed: Aug. 03, 2017]

[21] Hyeon-gu Lee, Minkyoung Kim, and Harksoo Kim, "Answer Snippet Retrieval for Question Answering of Medical Documents", Journal of KIISE, Vol. 43, No. 8, pp. 927-932, Aug. 2016.

[22] Liqiang Nie, Xiaochi Wei, Dongxiang Zhang, Xiang Wang, Zhipeng Gao, and Yi Yang, "Data-Driven Answer Selection in Community QA Systems", IEEE Transactions on Knowledge and Data Engineering, Vol. 29, No. 6, pp. 1186-1198, Jun. 2017.

[23] Kyoungman Bae, Youngjoong Ko, and Jonghoon Kim, "A Topic Classification Method Based on a Language Model Using the Structural Features of the Question-Answer Pair on cQA", Journal of KISS : Software and Applications, Vol. 39, No. 8, pp. 664-671, Aug. 2012.

[24] Jong Do Park, "A Study on Mapping Users Topic Interest for Question Routing for Community-based Q&A Service", Journal of the Korean Society for Information Management, Vol. 32, No. 3, pp. 397-412, Sep. 2015.

[25] Brent Hecht, Jaime Teevan, Meredith Ringel Morris, and Dan Liebling, "SearchBuddies: Bringing Search Engines into the Conversation", AAAI International Conference on Weblogs and Social Media, pp. 138-145, Jun. 2012.

[26] Kakao Talk Yellow ID API, https://yellowid.kakao.com/. [Accessed: Aug. 08, 2017]

[27] Amazon Web Services, http://aws.amazon.com. [Accessed: Aug. 08, 2017]

저자소개

김 보 경 (Bo-Kyung Kim)

2016년 3월 ~ 현재 : 금오공과대학교 컴퓨터공학과(공학사)
관심분야 : 딥러닝, 인공지능, 빅데이터

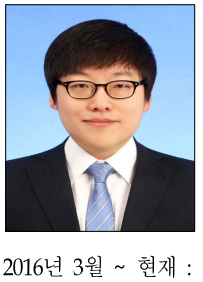
김 현 아 (Hyun-A Kim)

2013년 9월 : ㈜ CIG
2014년 10월 : ㈜ FUNFACTORY
2016년 3월 ~ 현재 : 금오공과대학교 컴퓨터공학과(공학사)
관심분야 : 게임

심 채 은 (Chae-Eun Sim)

2016년 3월 ~ 현재 : 금오공과대학교 컴퓨터공학과(공학사)
관심분야 : 인공지능

정 진 우 (Jin-Woo Jeong)



2006년 2월 : 한양대학교 컴퓨터공학과(공학사)
2008년 2월 : 한양대학교 컴퓨터공학과(공학석사)
2013년 8월 : 한양대학교 컴퓨터공학과(공학박사)
2013년 9월 : 삼성전자 책임연구원
2016년 3월 ~ 현재 : 금오공과대학교 컴퓨터공학과 교수
관심분야 : 멀티미디어 정보검색, 데이터마이닝, 사물인터넷 등.