[2페이지]

[3]페이지

팀원소개

[4페이지]

본격적인 프로젝트를 진행하기전 두 번 미팅을 하면서 팀원 소개 및 향후 챗봇제품 제작의 방향을 설정과, KT 챗봇을 보면서 챗봇의 구성이나 시나리오 등을 알 수 있었습니다.

KTDS 인턴을 시작하면서, 기존 챗봇 빌더나 오픈소스를 활용하여 어떤식으로 구성하는지 직관적으로 배웠고, 챗봇의 첫 단계인 의도분석 및 파악에 대한 작업을 수행했습니다.

[5페이지]

현재 메시징 앱의 확산으로 기업의 상품 및 서비스 제공 방식 또한 메신져 앱을 이용하는 추세가 되었습니다.

또한 AI 기술의 발전으로 언어의 맥락을 통한 사용자의 상황 분석이 어느정도 가능해졌습니다.

따라서 기업은 광고의 개인화가 가능해지고, 소비자들과 강한 유대감 형성이 가능해졌습니다.

또한 제품 문의, 환불 상담과 같은 업무에서 상담원이 퇴근한 시간에도 24시간 서비스를 제공 가능해졌습니다.

이런 배경을 바탕으로 케이톡, CJ대한통운 상담, 위비톡등 다양한 챗봇 서비스가 상용화 되어 있다.

기존 쳇봇 시장은 자연스러운 문장에 대한 정확한 답변이 어렵고, 정확한 패턴의 문장을 요구하거나 컴퓨터가 질의의 영역에 대한 선택지를 제공하는 룰기반으로 챗봇을 구성하였으나 현재 AI 기술이 발전 함으로써 자연어 처리를 바탕으로 정해진 패턴이 아닌 문장에도 정확한 답변을 주는 개발하고 있다.

우리는 특정 도메인에 한해서 구성되는 챗봇이 아니라 딥러닝 학습을 이하여 미사여구를 활용한 문장이나 구어체를 활용한 문장에 대해서도 의도파악과 감성분석을 통한

정확한 답변을 주는 챗봇을 기획하려 한다.

[6페이지]

챗봇이란 사람 대신 응답해주는 채팅 로봇. 즉, 사람과의 문자 또는 음성 대화를 통해 질문에 알맞은 답이나 각종 연관 정보를 제공하는 커뮤니케이션 소프트웨어이고 현재 콜센터, 물건주문, 호텔예약, 개인형 비서등 다양한 영역에 사용되고 있습니다.

사용자의 요청에 대한 의도를 파악하는 방식은 Menu\_Driven 방식과 AI\_based NLP 방식 두가지가 있는데 Menu\_Driven는 흔히 부르는 rule 기반의 챗봇으로 사용자에게 여러가지 의도를 제공하여 사용자가 의도를 직접 선택하게 만드는 방식이고

AI\_based NLP방식은 자연어 처리를 통해 컴퓨터가 직접 의도를 파악하여 답변을 제공하는 방식이다.

[6페이지]

머신러닝 방법중에 RNN이라는 모델이 있습니다.

RNN은 순차적인 정보를 처리하는 네트워크다.

‘recurrent’라는 용어는 모델이 입력 시퀀스의 각 인스턴스에 대해 같은 작업을 수행하고 아웃풋은 이전 연산및 결과에 의존적이라는 데에서 붙었다.

RNN은 이전 연산결과를 ‘기억’하고, 현재 연산 과정에서 이 정보를 활용한다는 점에서 음성인식, 이미지 캡셔닝, 자연어 처리와 같은 다양한 NLP작업에 사용되고 있는 모델입니다.

**Seq2seq는** 2개의 RNN을 연결하여 텍스트를 생성하는 모델로 입력과 출력이 같은 문장이 되도록 학습을 시키면 인코더의 출력인 벡터는 문장을 압축한 정보를 나타냅니다. 이 값의 코사인 유사도를 측정하여 두 문장이 얼마나 비슷한지 판별한다.

1.입력과 출력이 같은 문장이 되도록 사전에 학습을 진행합니다.

2. 그 후에 인코더의 출력인 벡터는 문장을 압축한 정보를 나타나게 되는데.

3.이 벡터 값의 코사인 유사도 측정 후,  두 문장의 유사성 판별

하는 방식으로 유사도를 측정할 수 있습니다.

[7페이지]

다음은 RiveScript와 Spring을 이용한 Rule 기반의 챗봇의 모습이다.

맨 오른쪽의 사진은 사용자가 “SAP 오류”에 대한 답을 정의하고,

유사 문장에 대비하여 “SAP 오류”의 뜻을 내포하는 문장들을 미리 정의하여 해당 문장으로 요청 받았을때 SAP 오류에 대한 답변을 제공하는 소스 이다.

가운데 사진처럼 “SAP 오류” 라고 입력하면 바로 답변을 받을 수 있다.

--> “SAP 오류” == “SAP 안되다” == “SAP 창이 하얗다＂ 모두 SAP 오류에 대한 답변을 제공해준다.

[8페이지]

Management : 단비에서 제공하는 관리 서비스를 이용하는 탭으로 의도추론, 사전등록, 트레이닝, 메신저 퍼블리싱 등 다양한 서비스를 이용할 수 있습니다.

Intent : 주어진 상황에 대한 의도를 설정하는 곳으로 하나의 의도 마다 하나의 chatflow를 구성할 수 있다. Intent안에 parameter를 넣어서 entity를 추출할 수 있다.

Chatflow list : chatflow의 목록을 볼 수 있다.

Chatflow는 주어진 질의에 대하여 답에 대한 시나리오를 구축할 수 있다.

Chatbot test : 의도파악은 잘 되었는지, 매게변수는 잘 전달되었는지 등을 서비스를 출시하기전 미리 알 수 있다.

[10페이지]

Chatfuel은 메뉴 기반의 페이스북 챗봇을 개발하는데 가장 적합하지 않을까 합니다. 무엇보다 사용법이 쉽고 무료이기 때문에 부담없이 개발할 수 있다.

Dialogflow는 다른 챗봇 플랫폼에 비해 기능이 약간 떨어지긴 하지만 무료라는 장점 때문에 많이 사용될 것 같다. 물론 초당 3개의 문장만 처리가 가능하다는 제한이 있지만 작은 규모의 챗봇이라면 큰 문제가 없을 듯 한다.

왓슨은 무료버전이 월 10,000개의 요청만 가능하기 때문에 테스트 용도가 아니면 사용할 수 없다. 하지만 기업 차원에서 챗봇을 만드는 것이라면 충분히 고려해 볼만한 플랫폼이라 생각한다.

단비는 국산 플랫폼이라 카카오톡을 지원하고 도움말이나 제품 지원이 잘 되어있다는 장점이 있습니다. 하루 10,000개 API로 왓슨보다는 무료버전의 제한이 덜한 편이지만, 실제로 챗봇을 운영하기 위해서는 유료 가입이 필요할 듯 한다

대중적으로 쓰이는 챗봇 빌더의 항목별 효율성을 정리한 챠트이다.

머신러닝을 활용한 자연어 처리분야에 대해서는 아직 완벽하게 처리를 하는 빌더는 없다 -> 어느정도 룰기반이 되어야 아직은 정확한 답을 전달 할 수 있다.

[10페이지]

의도분석에 대한 “각각 단어의 가중치를 설정하여 중심이 되는 단어를 찾아 의도를 파악하는 방식”, “비슷한 단어의 뜻을 유사어로 판별하여 문장의 유사도를 바탕으로 의도를 파악하는 방식” 두가지 방법이 도출되었습니다. 저희가 학생이고 이제 배우는 단계로 아직 둘 중 어느 방법을 써야 더 효율적인지 비교하며 사용해보고 싶었습니다.

따라서 의도분석에 대한 두가지 방법에 대해서 팀을 나누어 각각 방법에 적용하여 테스트를 진행한 후 더 정확한 방법은 무엇인지 보고 결정을 하고자 했습니다.

처음 의도 분석 방식은 단어의 가중치를 설정하여 중심이 되는 단어를 찾아 의도를 파악하는 방식입니다.

##Environment

파이썬 언어는 텐서플로우-케라스, 파이토치등 강력한 딥러닝 라이브러리가 존재하고 크롤링, 텍스트처리, 웹서버등의 강력한 라이브러리들을 손쉽게 사용할 수 있는 높은 생산성을 가지고 있다는 점에서 파이썬 언어를 주로 이용하게 되었습니다.

##의도분석1

자연어 처리에 있어서 워드 임베딩 기법을 적용하고자 하였습니다.

높은 학습비용으로 인해 미리 훈련된 데이터를 사용하는 것이 일반적이지만, 방대한 영어데이터에 비해 한국어는 데이터가 없거나 부적절한 데이터가 많아 직접 학습을 하였습니다.

위키백과를 백업한 덤프파일을 사용하였으며, 단어 간 유사도를 거리로 나타내었을 때 어느정도 만족할 만한 성능을 보여 주었습니다.

(우측그림)은 두 단어 사이의 코사인 거리를 나타낸 것인데 전혀 관련이 없는 '한양'과 '갤럭시'의 경우 멀리 떨어져 있습니다.

[11페이지]

한국어는 언어의 특성상 문장의 구조가 복잡하고 특히 어간 변화가 심해 전처리가 중요합니다.

따라서 저희는 문장을 형태소 단위로 분해하여 분석을 하였습니다.

khaiii, mecab, twitter등 다양한 형태소 분석기를 사용해 보았는데, 임베딩과 가장 잘 어울리는 분석기는 twitter 였습니다.

그런 다음, 원하는 핵심 품사들을 추출한 뒤, 임베딩하여, 모든 단어들을 100차원의 벡터로 바꾸어 주었고 그 과정에서 임베딩 사전에 없거나, 길이가 다른 문장들은 영벡터로 채워 넣었습니다.

[12페이지]

우선 테스트 데이터들을 만들었습니다.

문장 X는 클래스 Y1, Y2를 가지며,

Y1 = {“메일”, “페이먼트”}

Y2 = {“센드”, “리시브”} 이렇게

각 클래스는 2개의 라벨을 가지게 됩니다.

예를 들어, 0번 '메일을 보내고 싶어요'의 경우 “메일”과 “보내” 이 두단어를 캐치하여 Y1클래스는 메일, Y2클래스는 보내다 로 설정할 수 있습니다.

사진의 데이터로 분류기 모델을 간단히 테스트 하였습니다.

이중 테스트 데이터 중 가장 긴 문장은 6개의 형태소로 쪼개지고 그보다 형태소가 적은 문장은 패딩처리 합니다.

그런 후, 미리학습된 임베딩 레이어를 거쳐 RNN레이어, dense레이어를 지나 결과를 얻을 수 있습니다.

[13페이지]

테스트 데이터에 대한 분류는 성공적이었지만 적은 데이터로 train, dev, test로 나누지 않아 과최적화되어 있습니다.

그렇지만 모델이 나쁘지 않게 작동한다는 것을 알 수 있었습니다.

그러나 여러가지 문제점이 존재 했습니다.

첫번째로 문장 -> 라벨로 바로 잇는 모델로써는 라벨이 많아 질수록 더 많은 데이터를 필요로 할 가능성이있습니다.

이문제를 해결하기 위해서는 현재 정확한 데이터를 획득할 방법 필요한데 정확한 데이터를 획득하는 효율적인 방법을 모색하고 있습니다.

두번째로는 데이터가 존재하더라도 라벨링이 되어있지 않으면 라벨링에 매크로기반 챗봇을 개발하는 그 이상의 인력이 필요할 것입니다.

마지막으로 한국어의 문제점으로써 띄어쓰기를 하지 않거나 줄임말, 변형이 많고 그에 따라 임베딩에 한계가 존재하였습니다.

[14페이지]

다음은 문장의 유사도를 이용한 방법으로 의도분석 실험을 진행하였습니다.

문장의 유사도를 이용하는 방법은 FAQ와 같이 대화쌍 DB를 구축하여 사용자의 발화와 대화쌍 DB 간의 유사도를 계산하여 가장 유사한 대화의 답변을 반환해주는 방식으로 처음에는 이를 이용하여 기가지니 서비스 FAQ 챗봇을 구축하고자 했습니다.

저희는 파이썬 기반 KoNLPy라는 자연어 처리를 돕는 라이브러리와 mecab 형태소 분석기를 이용하였습니다.

우선 기가지니와 같은 고유명사를 미리 사전에 정의하고,

동의어의 관계망을 구축해야 하기 때문에 기가지니에 대한 뉴스파일을 크롤링시켜 word imbedding, word2vec 작업을 수행했습니다.

그 결과 기가지니는 인공지능, 음성이라는 단어와 거리가 가까웠고, 인공지능 스피커에 대한 유사어가 기가지니로 나왔습니다.

[15페이지]

하지만 앞서 사용한 기가지니를 중심으로 분석한 방법에는 “기가 지니”가 고유명사라는 점에서 유사어를 추출하기에 부적절했고, 부족한 데이터와 동사의 어미의 뜻이 변한 경우 인지를 하지 못한다는 문제점들이 발생하였습니다.

[16페이지]

이러한 문제점을 감안해 주제를 바꿔보았습니다.

이번에 정한 주제는 대학교 행정지원 도우미 챗봇입니다.

대학생활을 하다 보면 오늘 학식 메뉴, 행정실 전화번호와 위치, 교환학생 신청방법, 교내 장학, 건물위치 등 끊임없이 정보를 찾아야 하지만 전화로 물어보면 담당부서가 아니라고 전화가 돌려지고 통화중이거나 담당자가 자리를 비운 경우도 많은 등 평소에 필요성을 느껴 이기회에 행정지원 챗봇에 대한 FAQ로 테스트를 진행하였습니다.

[홈페이지를 통하여 정보를 찾을 수 있지만, 해당 메뉴와 정보를 스크롤하고 찾는데 오래 걸려 불편하기 때문에 학생들이 편리하도록 기존 메시징 어플을 활용하여 즉시 필요한 정보에 접근하는 방법인 챗봇 서비스로 제공하면 학생들의 만족도가 높을것이라 생각했습니다.](바쁘면 생략)

또한 필요한 데이터를 정확하게 선정하여 동의어 사전이나, 단어간 관계망 구축, FAQ를 쉽게 구축 가능하여 문장의 유사도를 통한 의도 분석이 가능할 것이라 생각했습니다.

학과 전화번호에 대한 문의를 할때 누군가는 “컴퓨터 공학과＂이라 하고, 또 다른 누군가는 “컴공“,”컴공과＂라고 하는데 우선 이러한 특정 학과를 지칭하는

줄임말들에 대한 유사성 테스트를 진행 하였습니다.

[17페이지]

우선 학과나, 행정지원 서비스에 대한 고유명사들을 사용자 사전에 정의했습니다.

그리고 각 학과를 지칭하는 유사언어를 임베딩하여 유사어간 관계망을 설정하였습니다.

그 후, (본 사진과 같이)test로 “컴공 전화본호”라는 대화를 입력했습니다.

형태소분석기를 통해 “컴공” / “전화본호”로 나누어지고 “컴공”과 가장 유사도가 높은 단어, “컴퓨터공학과“를 가져와 해당 학과에 대응하는 전화번호를 반환할 수 있습니다.

적은 데이터로 테스트를 하였지만 그 결과로, 단어간 유사도가 양호하다는 것으로 판단하였습니다.

[18페이지]

문장의 유사도를 분석하는 과정에서 Doc2Vec이라는 방법을 알아냈습니다.

Doc2Vec은 Word2Vec의 확장으로 의미공간에 document 의 위치좌표를 학습하는 방법이 제안됩니다. Doc2Vec은 여러 단어들이 모이는 문장을 하나의 document id로 놓고 이를 하나의 단어처럼 생각하여 모든 스냅샷에서 다른 단어들의 위치좌표와 함께 평균을 취하여 context vector를 만들고 그 이후 Word2Vec 과 같은 원리의 방식으로 문장간 관계망을 구축하는 방식입니다.

Word2Vec은 비슷한 문맥에서 나오는 단어를 비슷한 값을 가지는 벡터로 임베딩 하기 때문에 단어의 문맥상 의미를 담고 있으며 다양한 텍스트 마이닝 문제에 사용되었고 이를 이용한 Doc2Vec도 문맥의 의미를 담을 수 있어 글에 나타난 의도를 분석하는 문제에서 높은 성능을 나타냈고, 단순하게 문서에 등장한 모든 단어들의 Word2Vec을 이용한 벡터의 평균을 이용한 경우보다 문서 군집화와 분류 문제에서도 높은 성능을 나타내는 결과를 보여주었다는 서울대의 논문을 확인할 수 있었습니다.

향후 행정지원 관련 업무에 대한 데이터를 가지고 각각 Word2Vec를 이용하여 유사언어들의 관계 학습을 시키고,

FAQ 대화쌍들을 훈련데이터를 활용하여 Doc2Vec을 통해 문장의 유사도를 추출해볼 계획입니다.

[19페이지]

감정분석에 따른 응답의 융통성이 진행된다. -> 자연스러운 대화를 주도

[20페이지]

20) 이 모든 기능을 가진 챗봇을 개발하기 위하여, 제일 먼저 해야 할 것은 메신저 API를 사용하여 테스트 환경을 구축하는 것입니다. 현재 UI를 구현하는것은 시간과 능력이 과도하게 필요하며, 이를 현재 통용되는 메신저 API로 대체하여 개발 환경을 만들고 그후 최종단계로 고유 UI를 개발하는것을 목표로 합니다.

그 후, 챗봇의 Baseline모델을 위하여 룰 기반을 설정합니다. 이는 챗봇의 고유 학습보다는 기본적으로 개발자가 정해놓은 툴에 따라서 특정단어에만 반응하는 챗봇으로 데이터 셋 학습이나 개발상의 오류를 잡기 위함입니다. 또한, 머신러닝을 통하여 학습하기전의 베이스 데이터로써도 역할을 수행하게 됩니다.

세번째로는 본격적인 데이터 셋 수집/학습 입니다. 머신러닝을 위해서는 광범위의 데이터가 필수이며, 이를 통하여 조금씩 벡터를 수정하여 오류를 줄여나가야 합니다. 하지만 여기서 주의할점은 데이터셋의 정제입니다. 틀린정보나 챗봇의 주제와 관련이 부족한 데이터 셋을 사용할 경우, 챗봇자체에서 학습이 잘못되어 왜곡된 정보가 답으로 나오게 됩니다. 이는 치명적으로써 이를 회피하기 위하여 데이터를 먼저 걸러 낼 수 있는 검열 방법이 필요합니다.

그 뒤에는 인공지능 모델을 설계하며 추가기능을 구현하는 단계입니다. 인공지능 모델은 앞써 말씀드린 의도분석과 감정파악을 위하여 동작하며, 자세한 방법은 앞에서 말씀드린 대로입니다. 추가 기능이라 함은, 단순히 사용자의 입력에 따라 기존데이터에서 답변하는 것 외에도 검색엔진 기능을 탑재하여 검색으로 원하는 답을 찾는다던가, 사용자의 감정에 맞추어서 같은 질문에도 다른 답변을 제시할 수 있는 기능들을 뜻합니다.

마지막으로, 모든 기능을 구현한 뒤 이를 자체 웹 어플리케이션으로 전환해야 합니다. 디자인 기능은 너무 지루하지 않게 해야하며, 또한 복잡할 경우 접근성이 떨어지기 때문에 쉬우면서도 단조롭지 않은 디자인을 목표로 해야합니다. 또한 친근한 이미지를 설정해야 호기심에 사용한 유저들의 발걸음을 추후에도 다시 불러모을 수 있습니다.