9/14~

최종방향성

1.질문핵심파악(토픽모델링)

2.답변중에 어투가 이상한것들에대한 대체 질문(추천)

3.답변이 잘 나왔는데, 이와 관련된 질문을 물어봄

[것을 어떻게 설계하고 모델링할것인가..

bertModel을 구성하는 모듈이 하는역할]

일단이건 fix

최종해결목표 -> 챗봇의 질문의 답변이 hallucination 인지 아닌지 + 추가적으로 학습된 내부 DB를 참조했다면, 어디를 참조했는가 에 대하여

그렇다면 NLP 적 관점에서 보았을 때, hallucination이 무엇인지, 이게 어떤 이유에서 발생하는지, 이것과 정상적인 답변을 구분하는 해결점은 무엇인지

오픈소스 파인튜닝 끝~? [큰 방향성]

1. 임베딩에서 hallucination 판단열쇠가 있나?

-논문읽고 이해 / 정리 할 내용있으면 정리

-임베딩을 해서 고차원벡터로 변환되었을 때,

-일단 QA챗봇이므로 웬만하면 DB(매뉴얼 문서)를 참조해서 나오는 형식이면 좋은거니까, 그렇게 할 수 있는 방법과

-

1. Kochat 들여다보고 시스템 이해해보기(이거는 좀 별로인듯)

<https://github.com/hyunwoongko/kochat/blob/master/docs/02_about_chatbot.md>

1. Langchain 강의 및 티스토리 참고해서 코드 테스트 해보며, 원리 파악(db에서 끌어와서 쓰는[\*\*이걸 뭐라하지?->DB참조?] (수요일 수행예저)

-코드는 짜놓음, 이 세부적인 테스크를 이해해보려고 함

3.softGpt가 모델을 파인튜닝하는 식이 아니기 때문에 프롬프트나 그외의 것을 어떻게 잘 변경 시켜야 hallucination이 안일어날수있을까?

4.Hallucination이 발생하는 근본적 원인에대해서 깊은 탐구(논문읽기)

5.학습된 DB를 참조할수있으면, 우선적으로 참조하는 방법(이건 문서를 참조하는 QA봇 코드를 공부해보면서 어떻게 하면 우선참조 가능할까 연구[Langchain])

9/18~

[월요일]<vicuna, korani써서 프롬프트, input 파인튜닝해서 deploy해보기>

\*\*프롬프트 구조:

PROMPT = """우리는 아래와 같은 정보를 갖고 있습니다.

---------------------

{context}

---------------------

### 주어진 정보에 따라, 질문에 답해주세요.: '{question}'

### Assistant:"""

\*\* 프롬프트 테스트 + Testcase짜기

[화요일]

\*\*실패

\*\*Hugg ace api다운로드(모델 크기가 너무커서 api형식으로는 불가능하였음)

\*\*어떻게 이걸테스트 하는지 잘 모르겠음 자꾸 런타임 에러 뜨고, 테스트,deploy어떻게 해야할지모르겠음(실패)

\*\*논문읽기전 reasoning, hallucination, Interactivity 에 대한 search

[수요일,목,(금)]

\*\*\*\*논문정리 (이해안되서 다시 기본개념부터 좀보기) [목요링1번]



\*\* [롬프트 엔지니어링]

------------------

\*\* 문서를 기반으로 대답하는 챗봇 유투브,자료 참조해서 주석달고 깃헙 올리기

\*\* sentence\_embedding 코드 보고 이해

\*\* gpu 사용방식과 cuda 및 gpu를 쓰면 뭐가좋은지 개념혼동 다시찾아보기

\*\*비쿠냐 gpu들어오면 돌려보기 (협력)

\*\* 감정기반 분류로 kobert 어떻게 쓰고, 임베딩 어떻게 되는지 알아보기 코드테스트(해보고싶었다)

\*\* pre\_trained모델을 어떻게 사용하는지, 상세 방법 tokenizer, (encoding)코드화

\*\* 프롬프트에 ‘이렇게이렇게 나오게 해줘’ 라고 해서 다 그대로 나오는게 아니다. 그것을 정교화 할 수 있는 것을 찾아내야한다?

-금요일

\*\*LLM hallucination관련 논문읽기(위의 정답에 답이 될수있들만한 것 체크하기)

<\*\*\*\* 그렇게 하기위해 문서기반 챗봇을 일단 만들어보고 그 과정에서 해답을 얻어보자>

[사용기술]예상

-langchain의 openai 임베딩, 리트리버 QA, openai llm(chatgpt 3.5 turbo), chromaD

갤러리 참조

문서를 불러오기(Data loader)

헷 api 에 입력토큰수 맞추기

임베딩 방식과 문서를 벡터DB에 저장(chroma DB)

이 벡터 DB를 기반으로 검색해주는 retriever설정 vectordb.as\_retriver

체인만들기(검색+답변생성)

설정된 파라미터들

<Llm, chaintype, 리트리버, source\_document->출처표시 true>

+일단 영어로 질답이 되어있는데 한글로 되는지, txt파일인데 pdf도 되는지 추가적으로 테스트 해보기 (될거같음)

어떻게 문서기반 QA챗봇이 이루어지는지 과정을 알긴알았고, 여러가지 메소드,모듈라이브러리 등을 알았다.

\*\*메소드들 어떻게 구성되어있는지 알아보기 langchain(이게 결론적 해결에 의미가 있을지는 싶다)

\*\*여기서 hallucination test -> 문서에 없는 질문이나, 대답할 수 없는 형식으로 질문을 해보면 얘는 어떻게 반응할까

\*\* 이를 해결할 수 있는 방법은 뭘까

\*\*

근데 이 임베딩과 리트리버 모듈의 내부를 볼수없으니…

이 과정속에서 이루어지는 hallucination을 어떻게 방지하지? ,출처 표시는 vectordb를 통해서 해해결할수있을 것 같다. -> 논문을 읽자 +

++문장과 문장간의 유사도를 판별하는 실습을 어느 방법이라든지 해보자

++문장에서 핵심 context를 파악하는 실습을 해보자

\*\* 주는 일 ++

사용자 질문임베딩

영어 트랜스래이션

모델갈아끼워보고, 잘안되면 케이스

Use question encoder

필터링

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

사용자 질문에서 어떻게 답을 찾는가

qa

\*\*월 문서기반 답변 QA 챗봇 결과가 어떻게 나왔으며

검색기를 사용해서 llm\_response가 나왔는데, 결과를 일단 잘 해석해보고 이렇게 답은 잘나온다 식으로

이런걸하는이유, 질답과정에서 hallucination이 안생기고, 출처를 분명히 할수있게 하는게 최종목표인데, 이걸 구현하기 위해서는 그이전의 과정들도 필요하기 때문에 QA질답의 전체적인 과정을 이해해보려고 계속 llama,langchain,openai같은 오픈소스에서 코드를 찾아서 테스트를 해보고있는중이다. 그리고 이런 일련의 과정말고, llm(kobert,vicuna,)를 이용한 테스트도 해볼예정이다.

\*\* 문서데이터만 넣은 것 말고, QA데이터셋을 추가로 학습시킨 것을 합쳐서 나오게

\*\* 논문제대로 파악해서 위에서 계속 질문해오던것들을 답할수있게

\*\* 프롬프트입력- > 출력 잘보기 hallucination없나 있나 그 기준이 뭔가[앞으로의 위에서 제시한 해답이나, 방향성 찾아보기]

Test\_case

gp t를 통해 생성된 정보를 바탕으로한 정보를 통해 나오는 답변이 정확하게 prompt기반된 답변인지(G P T 3.5) [vicuna 33b 테스트기로 프롬프트 테스트]

--수욜 9/27

\*\*Hallucination 논문 의문점 및 정리

해소되지않은 질문: chatgpt같은 경우 입력창에 context가 아닌 질문만 입력하면, 답변이 형식과 질문의 의도에 맞게 잘나오는데, 이런경우 질문이 어떤 질문이 나올지 모르는데, context를 그에 따라서 어떻게 설정하는지, 아니면 질문 답변 쌍을 훈련을 시켜서 질문만 넣으면 답이 나오는건가? // 그러니까 context가 미리 결정되어있지 않고 질문만 넣으면 context를 설정해서 답변이 잘나오게 하는 방법,

이논문을 우리gpt에는 어떻게 적용할수있을까 내가 생각

우선 컨택스트가 참조하는 문서(manual DB,질문 입력->컨텍스트 결정(리트리버 사용?(이 질문에 맞는 컨텍스트를 매뉴얼 DB나 질답 데이터 셋에서 검색할때, 코사인 유사도나, 군집거리 등으로 제일 질문을 잘 참조할수있는 것을 찾음, DB 나 컨텍스트에 태그를 부여해서()context에 해당하는 부분에 태그)->프롬프트 완성->질답 데이터셋과 매뉴얼 DB가 pretrained 된 언어모델(Bert,gpt,vicuna)에 입력-> 추론->디코딩 및 답변출력 이런 형식으로 나오는게 맞는지 ,

\*\*Sql generation이나 다른업무를 수행하기 위해 인코더를 만들고, 임베딩을 어떻게 할지 아직 결정이 안난 상태에서 코드를 구현해야 하는 목표가 있는 것 같다. 따라서 여러가지 NLP 모델 테스크 들을 해보면서 감각을 익힐예정 –Bert/KoBert

-KoBert -> 한국어 위키백과 문장 5백만개, 뉴스기사 2천만개 학습 LLM 모델

-자연어 문장의 감정 분류

-

\*\* 논문에서 언급된 태그 방식을 사용하는 방법을 하려면 질문이 참조하는 컨택스트가 참조하는 문서(매뉴얼 DB, QA데이터셋

\*\*10/4 ~

1순위: 주시는 업무

-코드 메소드 상세분석

<langchain 사용>

{\*\*문서에 있는 article내용을 context로써 사용하기 위해, LLM 모델의 context length 에 초과되지 않게 순서대로 context text를 분류 해서 documents변수에 리스트 형식으로 넣어준다

\*\*vector DB 사용(from langchain.vectorstores import Chroma) <https://velog.io/@tura/vector-databases> ->sqlite파일로 vector화된 데이터가 DB로 저장되는데 이 DB안에 있는 내용을 확인하여서, 어떤데이터가 들어갔는지, 어떻게 임베딩ㅇ됬는지 확인

\*\*Chroma 에서 .as\_retreive , as\_relevant\_document 등을 사용해 vectorDB내 주어진 질문에 대답할수있는 답변검색

}

\*\*답변이 여러 개의 문서를 참조해서 잘 나오는지, 원리가 뭔지 어떤 프롬프트 형식을 쓰면 이렇게 되는지 (langchain 라이브러리 )

\*\*문서가 참조된 부분이 참조된 방식(어떤식으로 참조됬는지, 참조 정도는 어느정도인지, -> 직접 예시를 들어서

+RAG 와 retreiver에 대해서 좀 research + transformer에대한 상세한 이해 해보기 -> context의 선택 방식에 Rag 나 retreiver의 역할이 크기 때문

+이게 어느 알고리즘을 사용되면 이렇게 될까 (코사인 유사도, 맨해튼 거리 유클리드 거리,

+직접 코드 좀 짜보기 retreive코드참조

\*\*답변 분석

답변:”what is generative AI?”

참조 context :서로다른 3개의 문서

->어떻게 참조하는지 모르겠어서 문장 문단, or 문장 문장의 유사도를 구하는 것을 코드로 구현해보고자 함

-임베딩

10/5

답변이 잘 나왔다고 가정하고, 이와 관련된 질문을 물어봄 ->->어떻게 실행시킬지 방향성

-방법론적 방향-

기본 토대: 질문 질문 유사도 konlpy , okt 로 한글 형태소 단위로 토크나이징을 한다

Try A. 원래의 질문 셋을 vectorDB(weviate, ChromaDB)등에 임베딩후 집어넣고, 사용자가 질문을 했을 때, retriever query 에 그 질문을 집어넣고 검색하면, 벡터 거리 기반으로 유사한 질문이 나올것같다.

Try B. word2vec, Tf-Idf 등의 벡터화 방식 사용해서 유사도 거리 구하기, 거리가 가까운순 sorting 하는 방식으로

Try C. LLM의 토크나이저,임베딩 사용

-유사 케이스 조사(은행 챗봇 위주)-

기업이나 은행 챗봇의 경우 질문 종류가 자신이 물어보려는 키워드 중심의 질문이 많았고, 추천질문 역시 물어보는 목적어가 되는 키워드를 포함하는 질문이 많아서, 문장 벡터화와 유사도 측정 중점을 핵심 단어를 파악하는 것이 중요, 추천질문중 겹치는 단어가 있다고 해서, 그둘이 유사한가는 생각해봐야할듯

1. Langchain 코드 따라가 보면서 프롬프트가 어떻게 구성되어있는지, response가 어떤방식으로 나오는 지 따라가보기[너무 얽혀있고, 프롬프트가 제대로 안나와있는것같아서 이건 일단 보류하고 -> (llm\_response from multic docs) 이거에 대해 search

<https://medium.com/@tushitdavergtu/multiple-document-summary-and-llm-powered-qa-system-9b4df618c22>

<https://betterprogramming.pub/building-a-multi-document-reader-and-chatbot-with-langchain-and-chatgpt-d1864d47e339>

-langchain이 쿼리(+프롬프트) 가 구성되는 그 전까지의 과정

제일중요해보이는 QA chain이 형성된 근간이 되는 vectordb를 만든 Chroma 객체 ->langchain.vectorstores 🡪 vectorDB생성 = 사용자가 쿼리(질의)를 하면 벡터 DB에 분포값과 가장 가까운 벡터거리 순으로 질의와 가장 가까운값을 return해주는 형식

from langchain.docstore.document import Document

from langchain.schema.embeddings import Embeddings

from langchain.schema.vectorstore import VectorStore

from langchain.utils import xor\_args

from langchain.vectorstores.utils import maximal\_marginal\_relevance

1. 한글-영어 번역 모델 deploy

-vicuna korani move +프롬프트 엔지니어링 <git>

10/6 [일단은 한글->한글 방식의 일반적인 질답형태에서의 질문 추천을 고려해볼생각]

4한글-영어 모델 deploy

1sql모델 사용해보기 (아침에 말하기)

0 여러 개의 추천질문 뽑는 것(tf-idf방식 , 그리고 이 방식에 대한 이해) , 매뉴얼 QA csv좀 자세히 봐보기

~~Try A. 원래의 질문 셋을 vectorDB(weviate, ChromaDB)등에 임베딩후 집어넣고, 사용자가 질문을 했을 때, retriever query 에 그 질문을 집어넣고 검색하면, 벡터 거리 기반으로 유사한 질문이 나올것같다.~~

Try B(clear). word2vec, Tf-Idf 등의 벡터화 방식 사용해서 유사도 거리 구하기, 거리가 가까운순 sorting 하는 방식으로(지금해보는중)

-Tf-Idf 벡터화 방식 해봄 ->결과 잘나옴, 임베딩된 결과로 벡터거리를 계산해서 가장 가까운 질문을 뽑아봤는데 그럴듯하게 잘 추천했다.(0)

<근데 이렇게 형태소로 토크나이징 후 벡터화해서 거리 유사도 로 구별하는 것은 차이가 없어보인다 (잘나오는거 같긴함, 추천해주는 질문들이 타겟 질문과 상당히 유사하고, 특히 나아가서 물어봐야 할거라든가, 이 타겟질문내용 안에서 모를수도 있는 내용을 추천질문추천 추천해주는 것 같다)>

~~Try B2. 최근접 이웃 협업 필터링 사용해보기~~

~~Try C. LLM의 토크나이저,임베딩 사용(Kobert 사용해볼 예정)~~

-Llm 토크나이징/임베딩 후 임베딩된 벡터를 반환받는법 알아보기

~~1다른방식의 임베딩, 유사도 구하는 방법알아보기,~~

10/10

[2추천시스템 쪽 으로 공부해보기

\*\* gpt 프롬프트를 이용해서 질문추천해보기(프롬프트 안에 타겟 질문에 나아가서 물어봐야 할 것, 또는 이 타겟질문내용 안에서 질문자가 모를수도 있는 내용을 추천해달라는 내용을 추가 하는 식으로 해볼까함)

1. 문장,문단,문서의 키워드(핵심 부분 찾는부분에 대한 넓은 의미의 의미파악)

문장Case. 짧은문장에서의 키워드 , 긴문장에서의 키워드, 서로다른 문장들에 대해서 키워드를 다르게 찾나 테스트

키워드Case. 키워드가 짧은 단어로 나오는지, 어떤기준으로 키워드를 뽑는지, <이 키워드를 어디에다 쓰냐 에 따라 방향성이 결정될거 같다( ) ->(softgpt기준) 질문을 매뉴얼DB 와 질답 DB 에서 rag한 context가 있다. 이 질문과 context가 모여서 프롬프트가 완성되고,

?키워드를 그럼 왜 뽑는거지(궁금한거 또 질문하기)

키워드를 뽑아내는 방법(단순한 생각)

-단순히 어느정도 긴문장일 경우 단어의 빈출정도가 높은 것을 뽑아낸다.<단어의 의미를 고려하지 못할수도있으므로 좋은 방법이 아닌거같다>

-단어들을 우선적으로 뽑은뒤 단어들간의 유사도를 비교해서 제일 높은 것을 뽑아낸다.

1.토픽모델링 이해

LSA ->잠재의미분석(SVD에 대한 이해 필요)

LDA

SVD->mxn 직사각 대각행렬, ] ->일단보류

-프로젝트 차원 시키신일 -> 1. 논문 코드부분에 대해서 그 주변 맥락읽고, 코드가 무슨 흐름인지 알아보기

2. 보내주신 사이트 이해하기

코드&논문 \

* 논문에서 Experiments & data Collection부분

실험결과를 체계적으로 분석하기 위해~ 구조화된 JSON파일에 다양한 결과를 저장한거다

어떻게 태그를 붙였는지, 즉 실험준비 과정에 대한 언급이 없다.

<https://github.com/pgfeldman/KeywordExplorer/tree/main/keyword_explorer/experiments>

이주소에 ‘우리의 실험을 지원하는 모든 관련 코드는 GitHub에서 찾을수있습니다’ 라는 설명이 달렷다.

10/11/2023~[중간점검]

현재 해보자하는 방향성

NLP- sqlgeneration 이든 chatbot이든 아래것들은 필요하다

Rag

* Rag를 하는 시점?
* 사용자가 질문을 던지면 Retrieval 모델이 관련 정보를 검색하고, 이 정보를 Generation 모델에게 전달하여 더 정확하고 의미 있는 답변을 생성하게 한다. 이러한 방식으로 RAG는 생성 모델만을 사용하는 것보다 더 많은 정보를 활용하여 더 유용한 텍스트를 생성할 수 있다. -> 그럼 관련정보를 어디에서 context?, DB? ->이거는 rag를 사용하는 예시 코드에서 알아보자 sql+rag 코드 참조해보자(여기서부터 시작)

embedding해서 문단,문서,문장 의 topic 모델링

1. 문장,문단,문서의 키워드(핵심 부분 찾는부분에 대한 넓은 의미의 의미파악)

문장Case. 짧은문장에서의 키워드 , 긴문장에서의 키워드, 서로다른 문장들에 대해서 키워드를 다르게 찾나 테스트

키워드Case. 키워드가 짧은 단어로 나오는지, 어떤기준으로 키워드를 뽑는지, <이 키워드를 어디에다 쓰냐 에 따라 방향성이 결정될거 같다( ) ->(softgpt기준) 질문을 매뉴얼DB 와 질답 DB 에서 rag한 context가 있다. 이 질문과 context가 모여서 프롬프트가 완성되고,

?키워드를 그럼 왜 뽑는거지(궁금한거 또 질문하기)

키워드를 뽑아내는 방법(단순한 생각)

-단순히 어느정도 긴문장일 경우 단어의 빈출정도가 높은 것을 뽑아낸다.<단어의 의미를 고려하지 못할수도있으므로 좋은 방법이 아닌거같다>

-단어들을 우선적으로 뽑은뒤 단어들간의 유사도를 비교해서 제일 높은 것을 뽑아낸다.

1.토픽모델링 이해

LSA ->잠재의미분석(SVD에 대한 이해 필요)

LDA

SVD->mxn 직사각 대각행렬, ] ->일단보류

2.사용가능할만한 모델 -> top2vec <https://medium.com/@janhavil1202/understanding-topic-modeling-with-top2vec-cdf58bcd6c09>

추천시스템

[2추천시스템 쪽 으로 공부해보기

\*\* gpt 프롬프트를 이용해서 질문추천해보기(프롬프트 안에 타겟 질문에 나아가서 물어봐야 할 것, 또는 이 타겟질문내용 안에서 질문자가 모를수도 있는 내용을 추천해달라는 내용을 추가 하는 식으로 해볼까함)

hallucination 방지 source tag법 코드분석/이해\

코드&논문 \

result부분을 일단 다시 찬찬히 보면서 이 내용이 코드에 어떻게 구현되었는지 생각해본다

* 논문에서 Experiments & data Collection부분

실험결과를 체계적으로 분석하기 위해~ 구조화된 JSON파일에 다양한 결과를 저장한거다

어떻게 태그를 붙였는지, 즉 실험준비 과정에 대한 언급이 없다.

<https://github.com/pgfeldman/KeywordExplorer/tree/main/keyword_explorer/experiments>

이주소에 ‘우리의 실험을 지원하는 모든 관련 코드는 GitHub에서 찾을수있습니다’ 라는 설명이 달렷다

* 코드 이해

\*\*핵심이 되는 코드 키워드, URL(goodurl,badurl,no\_ctx\_url등), source, knowngood source

\*\*good source, bad source 를 평가하는 기준

\*\*wikidipia같은 문서를 요약할 때, topic modeling이 쓰이는것같은데 그쪽한번보기 topics/..

10/12

~~NLP레포 커밋~~

일단논문 끝까지 읽기

메모장 참고해서 코드 계속참조 확인하고,

위에 기준들 있으니까 그기준확인

Meta\_wrapping1,2 .py 가 결국엔 핵심이다 -> soucre를 특정하는 부분이 나옴

그리고 context를 결정하는 topics/top2vec 부분과

임베딩해주는 openAI/openAIcomms 부분 분석해보자

<근데 이걸 어떻게 다른 gpt나 sql generation하는 부분에 적용을 할까>

단지 태그를 달아서 이 응답이 해당 context와 source num이 같으면 같다고url을 반환하는코드인데, 지금 여기에 실험을 할 수 있는 데이터도 없고 해서, 이거를 따오기는 좀 힘들어보인다.

http 200 exception을 달아서 url이 유효한가 판단 good/bad

이 url이 응답으로 나오는 이유가 context나 prompt내에서 url을 출력하라고 지시 했기때문이지 않을까? , 그리고 이 url응답이 200이면 유효하고 good으로 판단한거였으면, 답변에 근거가되는 prompt url이 지 않을까

의문점:

* No-context-prompt가 context를 참조하지 않았기에 언어모델(LLM)의 고유한 지식을 바탕으로 대답하므로, 허위지식이나 부정확한 지식, 또는 질문의 의도와는 맞지 않는 지식을 내보내는 Hallucinaiton이 발생하는데, 여기서 LLM은 어떻게 고유 지식을 활용해서 사용자의 질문에 대한 답을 내놓는걸까?
* <https://aws.amazon.com/ko/what-is/large-language-model/>

LLM 트랜스포머는 워드 임베딩을 사용하여 인코더를 통해 텍스트를 숫자 표현으로 사전 처리하고 비슷한 의미를 가진 단어 및 구문의 문맥은 물론 품사와 같은 단어 간의 기타 관계를 이해할 수 있다. 그러면 LLM은 디코더를 통해 이러한 언어 지식을 적용하여 고유한 출력을 생성할 수 있다.

10/13

-한국어 -> 영어 번역 모델 deploy/test

Hugging face로 ko->en 번역을 지원하는 모델이 별로 없고 그중 BLEU값이 높은 것을 추려보았다

<model deploy as gradio pipeline & transformer library>

Gradio->파이썬 모델을 웹UI로 deploy할수있는 라이브러리

-1. https://huggingface.co/Helsinki-NLP/opus-mt-ko-en (gradio와 pipeline을 통한 deploy성공)->이거를 파인튜닝 한 버전들이 BLEU값이 좋았다.

- https://huggingface.co/inhee/opus-mt-ko-en-finetuned-ko-to-en5 <-Helsinki 파인튜닝한 버전

-2.그리고 opus mt 파인튜닝 말고

<https://huggingface.co/hyunseoki/ko-en-llama2-13b/tree/main> ->gradio space로

1. 라마 13b를 파인튜닝해서 번역기로 만든 모델도 찾아보았다. 이거는 일단 모델다운로드중 (완료)

근데 이게 py파일로 짜여진 코드가 안나와서 찾고있는데.. 모델디플로이가 처음이라 회사gpu환경서버 에서 테스트 해보고 싶은데, 깃헙 주소가 없어서 모르겠네.. ->알아보는중

*Zero-shot machine translation*

Main\_task: Gradio app으로 space를 만들어놓은걸로 deploy -> 한글 쿼리문을 영어로 바꾼뒤 depog-sql로 실행시켜봐서 sql문이 잘 생성 되는지 확인 해보기(이거 테스트)

<Benchmark>

BLEU<- 이거 내용정리

<https://wikidocs.net/31695>

c h r-f <-이거 내용정리

<기본적인결과 정리>

[Helsinki-ko-en 기본 , HelsinkiNLP.py]

-기본적으로 ‘주어’를 산정하지 않아도 I, you 등의 말하는 주체를 알아서 산정해서 테스트 결과를 내놓는다 ->필요없는 부분인것같다

->은행 이자가 얼마나 되는지 알려줘 : I want to know how much interest there is in the bank

-일단 단어별로 띄어쓰기를 해서 입력값을 집어넣는 것이 테스트결과가 훨씬 잘 나왔다

-> 은행이자가 얼마나 되는지알려줘 x 은행 이자 가 얼마나 되는지 알려줘 0

[Helsinki-finetuned-model, Hel\_finetuned.py]

-BLEU, c h r – f 값이 원본 Helsinki NLp모델보다 좋은 수치가 나온다.

-위와 똑같이 ‘은행 이자가 얼마나 되는지 알려줘’ 라고 물어보았을 때, I,you등의 주체를 알아서 산정하지 않고, 주어를 말해지 않았으므로 > Let me know how much interest is in the bank 훨씬 원본 번역과 비슷하게 결과가 나왔다.

[

이렇게 모델들을 알아보았는데, 일단 구글에 한글 -> 영어 번역 LLM쳤는데 잘 안나오고,

Llama나 koalpaca 같은 경우는 챗봇으로 활용되어서 한글을 영어로 번역하고, 질답형식으로 리턴을 출력하였다.

\*\*문제점 sql-generation 전체적 프레임워크에서 우선 사용자 입력을 한글로 받으면 영어로 리턴을 받아와야 되는데, gradio로 웹api로 deploy한거라 번역한 text값을 받아오는방법이 있어야 할거같다.(3.local로 받아오는 메소드 찾기)

\*\*1.llamafinetuned model deploy완성하기

\*\*2.테스트 데이터 더 돌려보고, 번역결과 sql-coder에 테스트한거 정리

\*\*4.

컨테이너,도커

gpu서버 컴퓨터와 똑 같은 상황을 내컴퓨터에 복제해서 좋은 cpu와 gpu를 붙일수있는환경

putty설정

ll

docker ps

-

doker images

doker run –it

42

51.8

5:41

10/17/2023

질문추천 알고리즘 설계

어쨌든 타겟 질문에 대한 다른 질문/답변 혹은 그 타겟질문에 대한 답변과의 연관성을 토대로 질문을 추천해야될것같다는 생각이든다. 그렇다면, 그 연관성에 대한 기준점을 어떻게 설정해야할까?

* 1. (타겟)질문 : 질문 (단순히 두문장에 대한 유사도 측면) <이건이미 코드로 구현해봄 tf-idf)>
* 2.타겟질문 : 다른질문에 대한 답변(왜나면, 타겟질문이 다른 질문에 대한 답변 내용과 유사한 측면이 있다면, 그 답변에 대한 질문이 타겟질문과 유사한 측면이 있다고 말할수 있기 때문에 추천하지 않아야 된다. 즉 동일한 내용이 반복되어 질문되어지지 않도록 하기위함) 타겟질문 : 다른질문답변 유사도 낮은순
* [단순 키워드 기반 유사도 뿐만 아니라 잠재적 토픽분석도 필요할거같아서, topic모델링 도 적용해볼까한다.]
* 3.타겟질문의 답변의 키워드중 이 타겟질문을 제시한사람이 답변을 보았을 때, 모를만한 키워드/토픽을 포함한 다른 질문을 추천해주거나, 그 키워드/토픽 을 설명하고 있는 답변의 질문을 추천

<공부하고 참조한 사이트> + 코드는 git commit

* <https://velog.io/@jaehyeong/Basic-NLP-sentence-transformers-%EB%9D%BC%EC%9D%B4%EB%B8%8C%EB%9F%AC%EB%A6%AC%EB%A5%BC-%ED%99%9C%EC%9A%A9%ED%95%9C-SBERT-%ED%95%99%EC%8A%B5-%EB%B0%A9%EB%B2%95>
* <https://wikidocs.net/24949>
* <https://bkshin.tistory.com/entry/NLP-9-%EC%BD%94%EC%82%AC%EC%9D%B8-%EC%9C%A0%EC%82%AC%EB%8F%84%EB%A5%BC-%ED%99%9C%EC%9A%A9%ED%95%9C-%EC%98%81%ED%99%94-%EC%B6%94%EC%B2%9C-%EC%8B%9C%EC%8A%A4%ED%85%9C?category=1097026>

10/18

앞서참고했던 페이지 외에, 추가적으로 추천시스템 전반에 대한 내용 공부.

코드 구현작업

1번은 이미 구현했고, 2번과 3번중, 3번 타겟질문에 대한 답변에서 핵심키워드들을 골라서 그 핵심키워드와 가장 유사한 질문을 추천하는 작업을 먼저 코드구현작업 하고있음, [진행중]

위 구현에 필요한 기술들-> 질답 DB를 보니 답변이 길다 따라서, 긴 문단에 대해서 핵심 키워드를 뽑아내는 작업과 이 뽑아낸 키워드와 다른 질문들과의 유사도를 구하는 알고리즘과, 타겟 질문을 한 사용자가 답변에서 모를만한 내용을 뽑아내는 것을 어떻게 알고리즘화 할것인지를 생각해야 할것같다

10/23/2023

+홈페이지 챗봇

어느방식이든간에 키워드는 뽑아 내야하기에, 일단 문장 or 문단에서 키워드를 뽑아내는 방식으로

일단 유사한 질문으로 갈 후속 질문 방향자체를 키워드 형식으로 뽑아내야할듯하다.

내가 앞으로 포함해야할 내용(ppt에서 제시된 요청사항)

타겟질문에 대해서 답변을 내놓고 나서, <답변 자체에 대해서 연관 키워드>를 뽑아내야될거 같은데

10/24

-상세 구현-

textRank 방법 이해/학습

\*\*코드적으로 우리 기능에 맞게 pagerank/textrank방법으로 구현

전범위 구현 중 -> 코드 구현완료

토크나이저-> 사용자 지정 토크나이저 함수 정의 konlpy.okt사용

데이터셋->JEUS질답db에서 answer부분 개행문자 삭제 및 전처리 후 사용, 및 명사만 추출

Tutorials/keyword\_keysentence\_extractor.ipynb 에서 코드 수정만 하면 될거같다-> 토크나이저랑 데이터셋 => 구현완료 but issue -> 중요 키워드가 너무 뻔한 답만 나오고, 조사 형용사들이 대거 등장한다 이부분 코드 수정이 필요하거나 다른 토크나이저를 사용해야할수도 있을것같다.

명사만 토크나이징 시켜서 이슈해결 하지만, 어느정도의 수준으로 잘 나오지만 이걸로, 추천시스템에 활용하기는 어려울것같아서, chatgptapi를 활용하는 방안도 고려해봐야할거같다.10/25 Chatgpt는 맥락을 고려한 핵심키워드를 잘뽑지만, co-occurancy 기반 단어 그래프로 키워드를 뽑는 것은

--------------------------------------------------------------------------------------------------------------

문단/문장에서 핵심 키워드 뽑느게 잘되야지만 추천 문제를 잘풀수있을것같다.

~~키워드 잘뽑히면 일단은 ppt요청사항은 충족시킨것같다.~~

Test.py에 토크나이저 랑 테스트 데이터셋 구해서 키워드 잘 뽑히는지 테스트해보자()

어떤기준으로 키워드를 추출했는가->간단하게 flow정리 10/25\*\*완료되면, 3번기준에 맞는 추천알고리즘 개발 세팅 빌드 해보기

타겟질문과 , 그에대한 답변에서

핵심문장을 뽑느냐,

핵심키워드를 뽑아서 (gpt유사도를 측정하느냐

10/25/2023

타겟질문을 한 사용자가 이후 궁금해할만한 후속질문 찾기

1. 우선 질문에서 사용자의 니즈를 파악하는 알고리즘 필요
2. [질문에 대한 답변에서 사용자가 후속으로 궁금해할만한걸 캐치하는 알고리즘필요

]>>이걸알기 위해선 사용자의 타겟질문의 키워드를 분석해야된다는 결론이 나옴 사용자의 지식수준이 질문을 어떻게 했는지, 즉 질문의 “맥락을 포함한 키워드에 나와있다는것이고 이것은 gpt로 찾아내는게 가장 결과가 잘나온다는 것을 여러 번의 키워드추출 테스트를 거쳐서 결론지었다.<결국 질문과 다른 질문의 유사도 높은순(이미구현) + 답변에서 타겟질문과의 유사도가 낮은<gpt로 타겟 질문과 그 답변의 키워드 비교해서 동일하지 않은 키워드를 핵심으로 잡고>-> 즉 타겟질문에서는 물어보지 못했던 키워드들과 유사도가 높은 다른 질문이나, 답변 에 매칭되는 질문을 후속질문으로 추천해주는 것이 핵심인데>

여기서 키워드 뽑는 건 이미 구했고(gpt api에 프롬프트만 살짝변경해서 사용 예정), 이 유사도 를 어떤 알고리즘으로 측정할지가->프롬프트에 context정보는 안넣는게 더 답변이 잘나오는것같다

<우선 추천 시스템 전반에 대해서 다시 복습 겸 공부 / tf-idf vectorization 상세>

10/26

사용자의 지식수준 -> 질문을 어떻게 했는지, 즉 질문의 “맥락을 포함한 키워드에 나와있다는것

사용자의 지식수준을 파악해야하는 이유 -> 그래야 사용자 지식 범위내에서 최적의 다음질문의 추천이 이어지니까, 추천시스템 알고리즘 최적화 의 핵심(내가생각하는)

사용자에 질문에서 추천질문으로 이어지는 요소가 뭘까

타겟질문의 키워드=사용자가 묻고싶은 것=사용자의 질문의도=지식수준

<gpt로 타겟 질문의 맥락을 포함한 키워드와 그 답변의 키워드 비교해서 동일하지 않은 키워드(유사도 가장 작은 하위2~3개 )를 핵심으로 잡고>

이거 코드구현 및 테스트 완료하기

<알고리즘 후보군> 이거 코드 구현 어떻게 할지

코사인유사도

k최근접이웃법

아니면 필터링?

10/27/2023

프로그램 캡슐화 작업, 및 파이썬파일 폴더정리

결국 추천은 두가지로 나뉜다.

1. 말그대로 질문과 유사한 질문->이거를 물어봤으면 이거도 궁금할걸? 이느낌[문장문장 유사도]
2. 질문에서 물어본 것외에 질문에 대한 답변에서 추가적인 궁금증 에 대한 질문->어 이 질문에 대해선 알겠어, 근데 이거를 이해했는데 이거에 답변에서 질문이랑 뭔가 관련이 있는거 같은데 이부분은 내가 지식이 없네?

🡺최종 알고리즘 flow

질문답변 db를 가지고 있다면 new질문에 대해서, 답변한 new답변, 거기서 일단 1차로 new질문 과 tf-idf score 벡터상으로 가까운 질문을 질문db에서 찾아서 추천해주고, 2차로 new질문,new답변의 핵심키워드를 추출한후, 그 키워드를 비교해서 동일한 키워드는 배제하고(답변의 핵심키워드가 질문의 핵심키워드와 겹치는 부분이 있다는 것은 그부분에 대한 설명이 이미 끝났다고 간주한다 따라서 그에대한 추가적인 질문은 없을것이고, 그 답변을 뽑아내는대에서 파생된 다른 핵심키워드 위주 즉 질문키워드와 겹치는 단어를 제외한 다른 키워드를 비교해, word2vec모델을 사용해서(실행시간을 줄이기위해 미리 모델을 학습시키고 load해놓음) 단어:단어 유사도중 가장 높은 상위n개의 키워드를 select, 10/30~) 11/02~ 마지막으로 이 n개의 키워드와 가장 score가 높은 답변(아직 어떻게 이 score를 구할지 못정함) 에 해당하는 질문을 생성하거나, db에서 가져오는것으로 구성된다

<이슈> 10/30~

~~키워드 들중 명사로 위장한 동사,형용사 조사 등이 있으므로 최상위 키워드 리스트를 뽑을 때, 이거를 거르는 알고리즘이 있어야할거같다~~ ->word2vec로 키워드 select할 때 =>굳이 거를필요없어보이는게 이미 gpt api를 거치면 명사만 추출되긴함

초기 질문에 대한 확장

중요한 개념 강조

다양한 측면 다루기

->이 3가지가 잘 지켜지는지 테스트 할 때 기준 생각해보기 10/30  
~

최종적으로 모든 함수 모듈화 종합 api화

10/31

Define\_recomm\_keyword\_test.py 에서의 에러(gpt api 에서 뽑은 키워드가 처음에 w2v에서 훈련시킨 토큰들과 맞지 않는 에러-> w2v의 메소드를 사용할 수 없는 에러 ->key error

1.공백으로 split해 원래 model에 학습시킨 토큰형태로 만들지 vs

2.오히려 이 단어를 모델에 새로 포함시켜(공백제거후) 학습시킬지->define\_recomm\_keyword를 두부분으로 쪼개서 일단 겹치는 키워드를 골라서 최종키워드를 선택한뒤, 이 최종키워드를 w2v모델에 토큰추가로 넣어 다시 모델을 정의 하고 원래 task대로 수행 (일단 이 방법이 그래도 내가 처음에 구상한 방법 => 근데 꼭 다시 모델을 정의 해야하나? , 내가 설정한 키워드만 추가적으로 vocab list에 더할순 없는건가 => 이거에 대해서 일단조사

\*\*\* 계속해서 키워드 : 키워드 유사도 비교해서 최종핵심키워드를 추출할 때 발생하는 문제-> 질답DB에ㅔ 있는 모든 문장들을 토크나이징 해서 w2v모델에 토큰으로 넣었는데, 누락되는 단어나, chat gpt api로 뽑은 키워드와 토큰으로 훈련시킨 w2v모델에 있는 vocabulary와 일치하지 않는 단어가 있으면 key error가 계속해서 발생한다 ->w2v 모델 vocabulary 리스트에 접근해 단어 토큰을 직접 추가하는식으로 해결했다

11/01

last step)

1. retreiver로 키워드가 포함된 답변 data검색해서 찾아내기

2. 임베딩을 해서 유사도 기반으로 거리가 가장 가까운 답변data찾아내기 ->

3. gpt api를 써서 prompt를 만들어서+ (질문)답변문맥 '이런 키워드를 잘설명해줄수있는 질문을 추천해줘 ' 이런식으로 질문 생성하기 테스트해보기 11/03

프롬프트 -> prompt=’답변[n]+’이 답변 문맥을 참고해서 다음 문단에서’+ final\_keyword\_list[n] +’와 관련된 추가질문을 만들어줘

4.예시 추천질문 보고 어떻게 해야할지

11/06~

-1차 selection 코드 수정하고 테스트완료하기 최상위 유사도 3개뽑을수있게

-Pr 이랑 issue 등록하기 NLP레포

-앞서 last step중 1개 알고리즘 flow 구체화

[나아가서 임베딩 모델 코드를 일반코드에 어떻게 적용할수있을지]

11/13/2023

word2vec 시간줄이는거 => 동음이의어는 못잡지만, 모델을 미리 학습시켰을때는 엄청빠름 ,but => 키워드를 추가하는 부분에서.. => 이것도 model 미리 save 해두고 로드만 해서 실행만 시키면 1초도 안걸림 // 해결

아키텍쳐 상에서 최대한 효율적이게 어떻게 추천시스템을 녹여낼까

우리시스템 flow

질문-리트리브 해서 답변db에서 불러온후 - gpt로 정제 느낌

은행쪽 DB => 질답 잘나와있고 , 간단한 질답형식일 듯

==> \*\*원래 내 알고리즘 + DB참조해서 이상한 질문이 들어와도 그 거랑 가장 잘맞는 질문 자동완성 알고리즘 기능을 할수있게

-> 이거는 1차 알고리즘에서 손봐야함

-new issue /task-

: 이상한 질문이 들어왔을때, 답변이 그 질문의 의도를 파악하지 못하고 답변을 해준다면, 후속질문으로 전 질문의 의도를 답변으로 반영해줄수있는 추천질문을 고려해야한다

-> 이 issue는 1차 에서 해결이 가능할것도 같아서, 1차에서 예시 케이스들을 테스트를 해보겠다

+ 추가적으로 1차 알고리즘의 score를 consine similarity로 구하는 법도 테스트해보겠다 -> 근데 굳이 안써도 될거같긴함 어짜피 벡터 계산으로 구하는것은 동일해서

+dense\_matrix vs sparse matrix 의 차이가 에러 해결의 핵심 key => 11/14 일단 에러자체는 해결을 했는데, 코사인 유사도로 구한 문장 by 문장 의 점수가 , 유클리디언 거리로 구한 점수보다 좀 안나오는 느낌이다.

->임의로 이상한 질문을 타겟질문으로 넣어놓고 1차 score를 돌렸는데, DB저장된 질문들중에서 그래도 꽤나 관련된 질문들을 잘 뽑아주긴한다.

+

11/15~11/21

~~1차 코드에서 좀더 신박하게 자동완성을 기능을 설명할 부분을 찾아보자 ㅇ~~

~~1,2,차 모두 api화 할수있게 코드 정리 ㅇ~~

~~NLP레포 readme 메모장 정리~~

\*\*논문 내용들과 내가 짠 코드들을 어떻게 연관지을수있을지

~~: 동음이의어를 word2vec 이 판별할때 예외 케이스를 설정해야할것같다.~~

11/22~

2차 코드 전체적으로 다시 훑어서 메인부분 강화+마지막 추천 리턴받을 때의 형식

Gpt api안쓰고 2번 방식으로 코드짜기/테스트 보기

문자와 문단의 연관성을 지표(score로 나타내는 방법에 대해 알아봐야 할거같은디)

1. **TF-IDF (Term Frequency-Inverse Document Frequency):** 이 방법은 문서 내에서 특정 단어의 중요성을 계산합니다. 특정 문자가 문단에서 얼마나 자주 나타나는지를 계산하여 해당 문자의 중요성을 평가할 수 있습니다.
2. **워드 임베딩(Word Embedding):** 단어를 벡터로 변환하여 의미적 유사성을 측정하는 방법입니다. Word2Vec, GloVe, FastText와 같은 워드 임베딩 모델은 단어 간의 의미적 유사성을 포착하므로, 특정 문자와 문단 간의 의미적 연관성을 측정하는 데 사용될 수 있습니다.
3. **N-gram 모델:** 특정 문자의 나열을 N-gram으로 나누어 문맥을 이해하는 방법입니다. 이를 통해 특정 문자가 문단에서 어떤 위치에 나타나는지와 관련된 정보를 얻을 수 있습니다.
4. **문장 임베딩(Sentence Embedding):** 문장 전체를 벡터로 표현하는 방법으로, 문장 간의 의미적 유사성을 평가할 수 있습니다.
5. **코사인 유사성(Cosine Similarity):** 벡터 표현을 사용하여 문자 또는 문단 간의 유사성을 계산하는 방법 중 하나입니다.
6. **Attention Mechanism:** 특정 문자나 문단에 더 중요한 가중치를 부여하는 어텐션 메커니즘은 특정 문자와 문단 간의 관련성을 모델링하는 데 사용될 수 있습니다.

1.이중 tf-idf 와 word2vec은 이미 테스트를 완료해봤는데 빈도수 기반이라 문맥을 파악 하지 못할거같아서, 문잭을 파악할수있고, 키워드를 강조하는 문단을 찾아야 하므로 임베딩 모델을 쓰거나, attention mechanism으로 문제를 풀려고 한다. 하지만 추천 알고리즘에서 모델을 쓰는 것은 너무 시스템이 무거워질수있으므로, 일단은 6번으로 알고리즘을 짜려고 한다.

15-01,16-02

근데 모델 들어가야되서 일단은 보류 (임베딩 모델을 쓴다고 해도, 어짜피 문자:문단 비교이므로 유사도 구분 성능이 확좋아질거같지 않다.

1. text-rank로 문단에서 핵심문장 뽑기(pagerank기반) // 이왕이면 키워드가 포함된 == 답변과 비교를 해야하는데 \*\*이슈 [DB답변 데이터를 전부 비교하기엔 핵심문장 뽑는과정이 너무 오래걸릴거같아서] => 이문제를 해결해야 할것같다 … -> 챕터에서 1차로 추리고, 그다음 비교해야할 듯

sentence embedding, word embedding

[<핵심문장>에서 키워드 뽑기 = 원래 뽑은거랑 비교

<핵심문장이 아닌>문장 = 원래 뽑은거랑 비교

Keywordrank랑 word2vec해서 키워드 뽑아야되고,]

[[system flow]]

1. user query날라옴
2. user query를 제일 비슷한 의도를 가진 질문DB의 question으로 대체(내부과정)
3. 그 q의 키워드를 가져옴
4. 답변의 키워드는 rag한 내용에서 가져옴?? \*\*이거질문 여기서 코드 돌아가는 동안에 gpt api를 안쓸수있나? (아니면 이것도 질문DB q의 해당되는 a로 대체?)
5. 키워드 중복삭제 및 핵심키워드 선택
6. 최종적으로 키워드 끼리 비교

11/27~12/08

12/05~

~~-csv에 잘들어가는지 for문 테스트만 끝내고,~~

-1차코드참조해서 질문을받고, db유사도 상위 질문으로 바꿔주는 코드 작성

-이렇게 되면 final score도 미리 저장해놓을수있을듯

<<구현방향>>

[미리해놓을거]

-git 관리(main 작업하고 master로 pr , merge)

-word2vec모델 학습시키는 코드 해놨다 )) word2vec\_test.py \*\*\*clear

-핵심문장을 합친 문단을 만드는 코드 작성(csv저장) \*\*\*clear

-문단DB 에서 키워드 뽑는 + full답변에서 키워드 뽑는 gpt프롬프트 코드<설명하는 키워드> 작성(gpt test)==>> 이런 flow면 미리 뽑아놓을수있음 [[ 미리 뽑아놓는 코드 작성]](csv저장)

컬럼10개로 임의 테스트 🡪 openai api키 필요 \*\*clear

-1차적으로 user q를 자동완성 기능으로 질문DB에 있는 질문으로 대체시킬수있는 코드작성

\*\*clear

-질문DB에 있는 질문들 키워드 뽑는 gpt 프롬프트 코드 작성(gpt test) 컬럼10개로 임의테스트 저장 (csv저장) \*\*clear

-최종 키워드 비교할 때 어떻게 비교할지 -> 다대다 vs 다 대 1\_\_ 이게 살짝 오래걸릴수도?

<2차 코드에 이어서 작성>

[다양한 분포를 가지는 점수들의 모임이 여러개가 있고, 그 모임들에서 가장 다양한 분포를 가지면서 총합이 높은 모임을 뽑으려고 하는데 이를 제일 잘 판별할수있는 수학적 지표]

1. **엔트로피 (Entropy):** 엔트로피는 분포의 불확실성을 나타내는 지표로, 값이 높을수록 다양성이 크다. 여러 분포의 엔트로피를 계산하고, 높은 엔트로피를 가진 분포를 선택할 수 있다.
2. **쿨백-라이블러 발산 (Kullback-Leibler Divergence):** 쿨백-라이블러 발산은 두 확률 분포 간의 차이를 측정하는 지표로, 값이 크면 두 분포가 다양하다고 판단할 수 있다.
3. **자세한 통계 지표:** 분포의 특성을 더 자세히 살펴볼 수 있는 통계 지표들도 고려할 수 있다. 예를 들어, 분산이나 표준 편차가 크면 다양성이 높다고 볼 수 있다.
4. **총합:** 분포의 총합 역시 중요한 지표다. 여러 분포의 총합을 계산하고, 다양성과 함께 고려하여 선택할 수 있다.
5. **다차원 스케일링 (Multidimensional Scaling):** 데이터의 차원을 축소하여 시각화하고, 분포 간의 유사성 및 차이를 살펴볼 수 있다.

**총합과 분산 표준편차를 더한 점수로 우선 총합 즉 유사도 점수 총합이 높은 것과, 분포가 다양한 유사도를 가진 것을 final 추천으로 select**

**0.7x총합 + 0.3x분산or표준편차 = final\_score**