



ARA

: Assistant of Research Analyst

Team MARA

Jonghwan Kim, Yikyung Kim, Jeongha Moon, Ji Hong Min, Hanjun Cho



SEOUL NATIONAL UNIVERSITY
GRADUATE SCHOOL OF DATA SCIENCE

Contents



- 1 Introduction**
- 2 Motivation**
- 3 Architecture**
- 4 Project Demo**
- 5 Conclusion**

1. Introduction



Team Members



Jeongha Moon
Project Owner



Yikyung Kim
Architecture



Ji Hong Min
UI/UX Designer



Jonghwan Kim
Development



Hanjun Cho
ML/AI Engineer

2. Motivation



Daily life of research analyst

- Monitoring news and events
- Conducting research on industries and companies
- Providing support to senior analyst and traders
- Preparing daily reports for clients

→ **Overtime is routine!**

*How can we enhance efficiency
by reducing simple working hours?*

2. Motivation



Instead of searching for all the articles



What about using NLP models?

A screenshot of a news website's homepage. At the top, there's a header for "Biotech And Pharma". Below it, the first article is titled "Biotech Stocks Are Heavily Shorted. They Could Be Winners." It includes a byline "By Jacob Sonenshine", the date "May 3, 2023 4:30 pm ET", and a summary: "Just under 45 million shares of the SPDR S&P Biotech ETF were sold short as of Monday, about 56% of the fund's share's up from about 30 million in August." There's a "Go to article >" button. The second article is titled "2 Biotech IPOs Are Coming. The Window Is Still Shut." It includes a byline "By Josh Nathan Kazis", the date "May 3, 2023 8:19 am ET", and a summary: "Two initial public offerings in the industry in a single week wouldn't have been big news in 2021, but this year it is enough to stop traffic." There's also a "Go to article >" button. On the right side of the page, there's a sidebar with tabs for "Change", "Health", and "Science". Under the "Health" tab, there are three news items: "r's progression by", "nurse staffing crisis:", and "medical errors kill". Each item has a small thumbnail image and a "Read more" link. At the bottom of the sidebar, there's a snippet from NBC News: "The pandemic worsened patient safety, with Leapfrog's new report showing increases in hospital-acquired infections. It's..." followed by a "Read more" link.

Please list and summarize the top-3 most important news in the healthcare sector in last 24 hours.



1. Biotech Stocks are Heavily Shorted.
2. The Biotech IPO Window isn't open yet
3. Pfizer Stock jumps after sales beat.

2. Motivation



Limitation of ChatGPT

- Unable to provide latest information

J please summarize the last night news in biology sector

 I'm sorry, but as an AI language model, I don't have real-time access to news updates or the ability to browse the internet. My training only goes up until September 2021, so I can't provide you with the latest news in the biology sector. I recommend checking reliable news sources or conducting an online search to get the most up-to-date information on recent developments in biology.

- Hallucinations

 조선왕조실록에 기록 된 세종대왕의 맥북프로 던짐 사건에 대해 알려줘

 세종대왕의 맥북프로 던짐 사건은 역사서적인 조선왕조실록에 기록된 일화로, 15세기 조선시대 세종대왕이 새로 개발한 훈민정음(한글)의 초고를 작성하던 중, 문서 작성 중단에 대한 담당자에게 분노하여 맥북프로와 함께 그를 방으로 던진 사건입니다.

세종대왕은 훈민정음을 개발을 위해 열심히 노력하던 중, 한글 문자를 이해하지 못하는 당시의 관료들이 문서 작성률을 높기 위해 임명한 기획관 중 한 명인 최환(崔煥)에게 작성 중단 사실을 들게 되었습니다. 세종대왕은 이에 분노하여 최환을 부르고, 맥북프로를 함께 들고 그를 방으로 불러들이며 화를 내었습니다. 이후, 세종대왕은 훈민정음의 초고 작성률을 다른 담당자에게 넘겨주게 되었습니다.

이 사건은 조선시대의 통치자로서 세종대왕이 어떻게 민족문화를 발전시키기 위해 노력했는지를 보여주는 일화로 유명합니다. 또한, 세종대왕의 열정과 업무처리에 대한 철저한 태도를 보여주는 사례로도 언급되고 있습니다.

Apparently, ChatGPT can't handle these critical issues. Can we come up with something new?

3. Architecture - Overview



What about using retrieval model?

ChatGPT

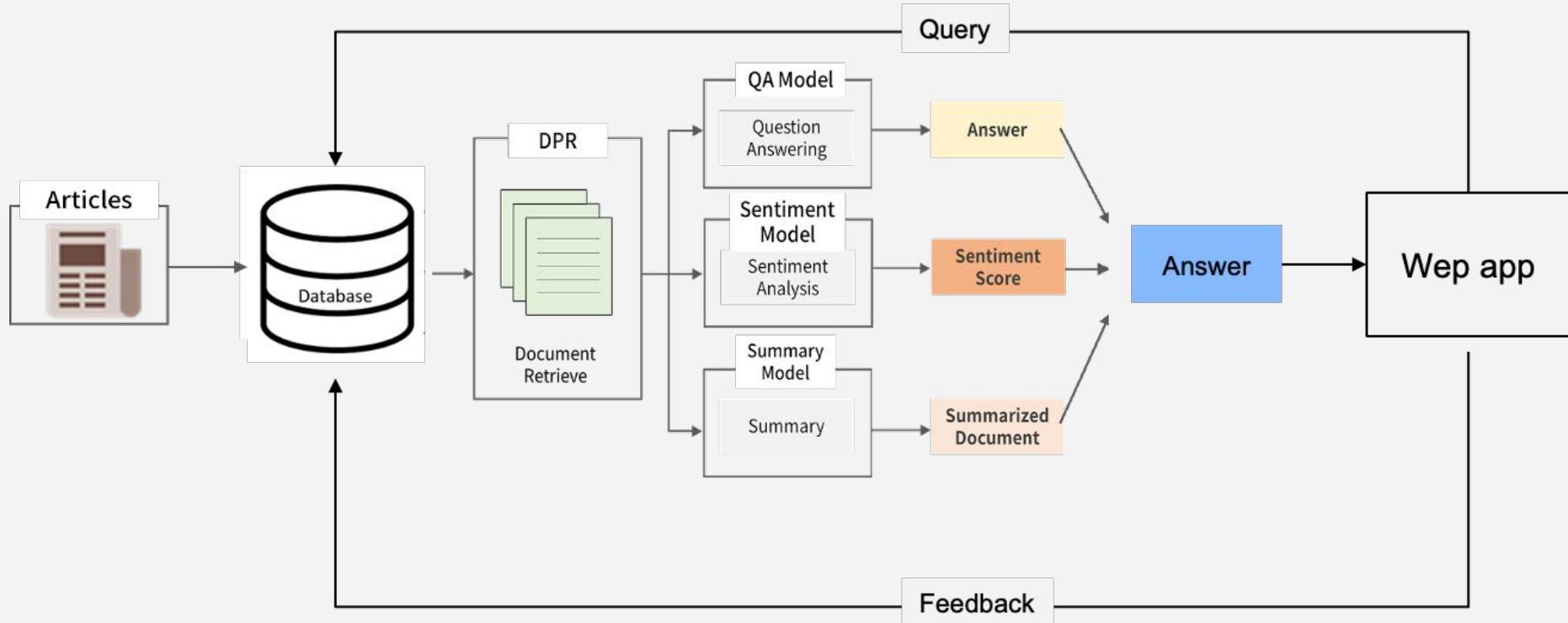
- 1. Generative model
 - : Creates hallucination problems
- 1. Pre-trained model
 - : Unable to provide latest information
 - : Power consumption is substantial (502 tons of carbon emission for training!)

Proposed Model

- : Retrieval + Generative model
 - 1. Generates based on the retrieved facts
 - > *No hallucination problem!*
 - 1. Retrieve facts based on the scraped data
 - > *Able to provide latest information!*

With this proposed model, ARA can provide both **timely and credible information** to users!

3. Architecture - Overview



3. Architecture - Data Collection



News Data Crawling



11 different sectors

Basic Materials
Communication Services
Consumer Cyclical
Consumer Defensive
Energy
Financial Services
Healthcare
Industrials
Real Estate
Technology
Utilities

yahoo!



mongo DB

```
{"id": ObjectId(2479274),  
"title": "Merck sues U.S.  
government over Medicare...",  
"provider": "Yahoo Finance Video",  
"url": "http://finance.yahoo.com...",  
"datePublished": "2023-06-13",  
"content": "Merck is suing the  
U.S. government..."}
```

3. Architecture - Data Collection

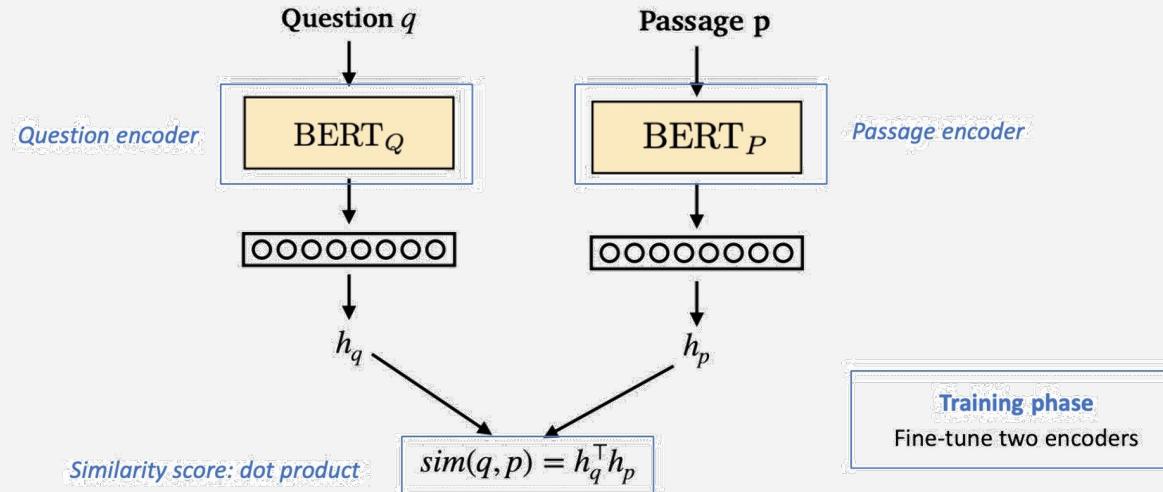
Batch Schedule : every morning @ 04:30 a.m

파일 형식	파일 크기	마지막으로 수정한 날짜	작성자	이름
.ipynb	43KB	2023. 6. 16. 나	나	Daily_job_schedule 2023-06-15T19:30:00Z.ipynb
.ipynb	43KB	2023. 6. 15. 나	나	Daily_job_schedule 2023-06-14T19:30:00Z.ipynb
.ipynb	42KB	2023. 6. 14. 나	나	Daily_job_schedule 2023-06-13T19:30:00Z.ipynb
.ipynb	42KB	2023. 6. 13. 나	나	Daily_job_schedule 2023-06-12T19:30:00Z.ipynb
.ipynb	49KB	2023. 6. 12. 나	나	Daily_job_schedule 2023-06-11T19:30:00Z.ipynb
.ipynb	45KB	2023. 6. 11. 나	나	Daily_job_schedule 2023-06-10T19:30:00Z.ipynb
.ipynb	49KB	2023. 6. 10. 나	나	Daily_job_schedule 2023-06-09T19:30:00Z.ipynb
.ipynb	45KB	2023. 6. 9. 나	나	Daily_job_schedule 2023-06-08T19:30:00Z.ipynb
.ipynb	43KB	2023. 6. 8. 나	나	Daily_job_schedule 2023-06-07T19:30:00Z.ipynb
.ipynb	49KB	2023. 6. 7. 나	나	Daily_job_schedule 2023-06-06T19:30:00Z.ipynb
.ipynb	43KB	2023. 6. 6. 나	나	Daily_job_schedule 2023-06-05T19:30:00Z.ipynb
.ipynb	43KB	2023. 6. 5. 나	나	Daily_job_schedule 2023-06-04T19:30:00Z.ipynb
.ipynb	43KB	2023. 6. 4. 나	나	Daily_job_schedule 2023-06-03T19:30:00Z.ipynb

3. Architecture - Model



Dense Passage Retriever



When a question comes in, top-k articles that are highly related to the question are retrieved

3. Architecture - Model

Dense Passage Retriever

```
import pymongo
db_url = 'mongodb://ara-prod:LDbvT8AQv0dptEbDX2GU8sAL5ogUzDqpdvCHPQJsfSMafLPqhWoLTxuY8DTEOPPMuCLw0MgyGczACDbUJztY==@ara-prod.mongo.co
DB_NAME = 'ARA_DB'
FEEDBACK_COLLECTION = 'FEEDBACK'
client = pymongo.MongoClient(db_url)
db = client[DB_NAME]
feedback_db = db[FEEDBACK_COLLECTION]
industries = ['ms_basic_materials', 'ms_communication_services', 'ms_consumer_cyclical', 'ms_consumer_defensive',
    'ms_energy', 'ms_financial_services', 'ms_healthcare', 'ms_industrials', 'ms_real_estate', 'ms_technology', 'ms_utilities']
ARTICLE_COLLECTION = industries[0].upper()
article_db = db[ARTICLE_COLLECTION]
articles = article_db.find({}, {'_id':0, 'content':1}) #content만 가져오기 (가져올 column : 1, 안 가져올 column : 0)
article_db.count_documents({}) #document 갯수 count
articles[0]
```

```
import time
import torch
from transformers import AutoTokenizer, AutoModel
from transformers import DPRQuestionEncoder, DPRQuestionEncoderTokenizer
from transformers import DPRContextEncoder, DPRContextEncoderTokenizer
from sentence_transformers import util

device = torch.device('cuda')

dpr_ctx_tokenizer = DPRContextEncoderTokenizer.from_pretrained("facebook/dpr-ctx_encoder-multiset-base")
dpr_ctx_encoder = DPRContextEncoder.from_pretrained("facebook/dpr-ctx_encoder-multiset-base").to(device)
dpr_query_tokenizer = DPRQuestionEncoderTokenizer.from_pretrained("facebook/dpr-question_encoder-multiset-base")
dpr_query_encoder = DPRQuestionEncoder.from_pretrained("facebook/dpr-question_encoder-multiset-base").to(device)
```

```
def select_candidates(query:str, sector:list, num=5) -> list:
    contexts = []
    for i in range(len(sector)):
        contexts.append(sector[i]['content'])

    with torch.no_grad():
        dpr_query_input = dpr_query_tokenizer(query, return_tensors='pt')[['input_ids']]
        dpr_query_emb = dpr_query_encoder(dpr_query_input.to(device)).pooler_output
        dpr_ctx_input = dpr_ctx_tokenizer(contexts, padding=True, truncation=True, return_tensors='pt')
        dpr_ctx_emb = dpr_ctx_encoder(**dpr_ctx_input.to(device)).pooler_output

        dpr_query_input.to('cpu')
        dpr_ctx_input.to('cpu')
        del dpr_query_input
        del dpr_ctx_input
        torch.cuda.empty_cache()

    cos_scores = util.pytorch_cos_sim(dpr_query_emb, dpr_ctx_emb)[0]
    top_results = torch.topk(cos_scores, k=num)

    top_k = top_results.indices.to('cpu')
    top_k = top_k.numpy()

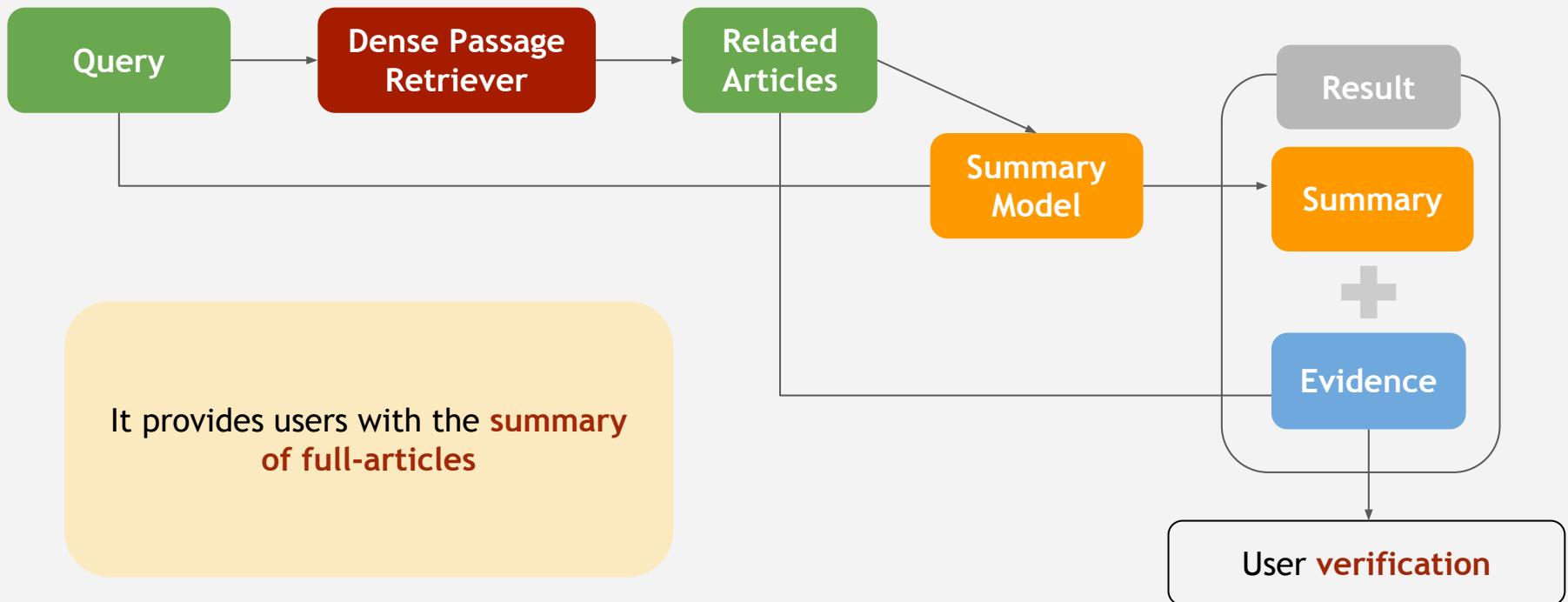
    L = []
    for i in top_k:
        L.append(sector[i]['content'])

    return L
```

3. Architecture - Model

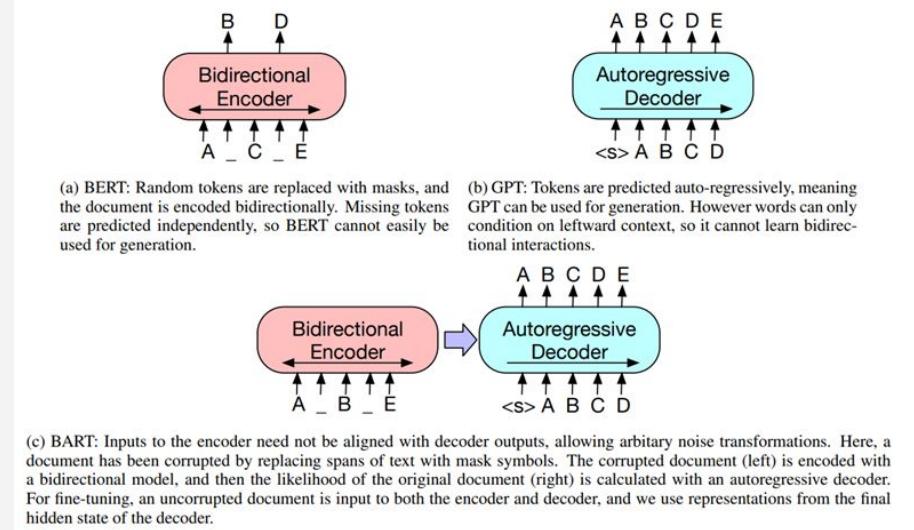
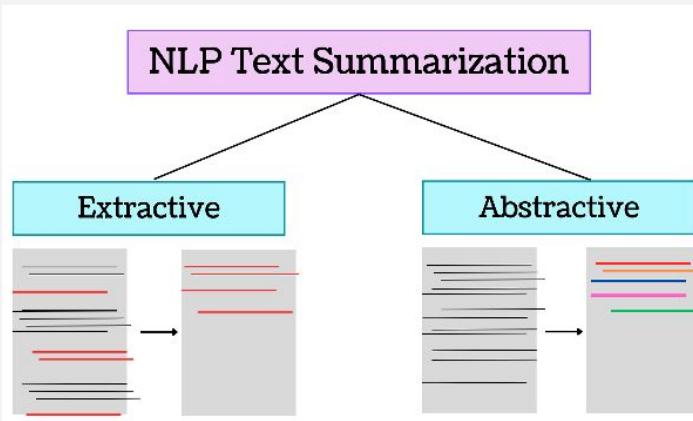


Summarization



3. Architecture - Model

Abstractive Summarization

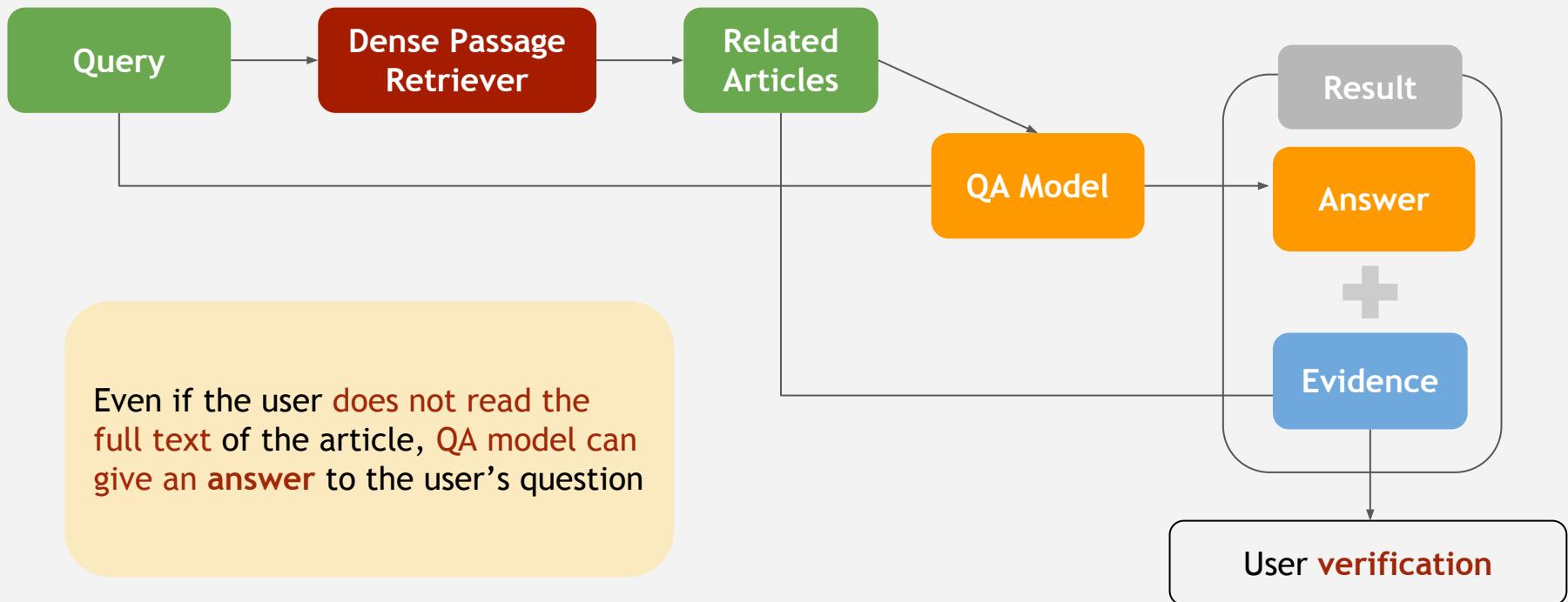


ARA adopts the Abstractive Summarization for fluent result

3. Architecture - Model



Question Answering



3. Architecture – Model

Extractive QA

[Example Question]

Is there a subscription fee for ChatGPT Plus?

Extractive: \$20

Abstractive: Yes, there is a \$20 subscription fee for ChatGPT Plus.

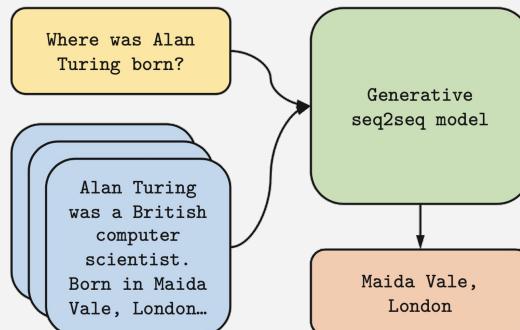
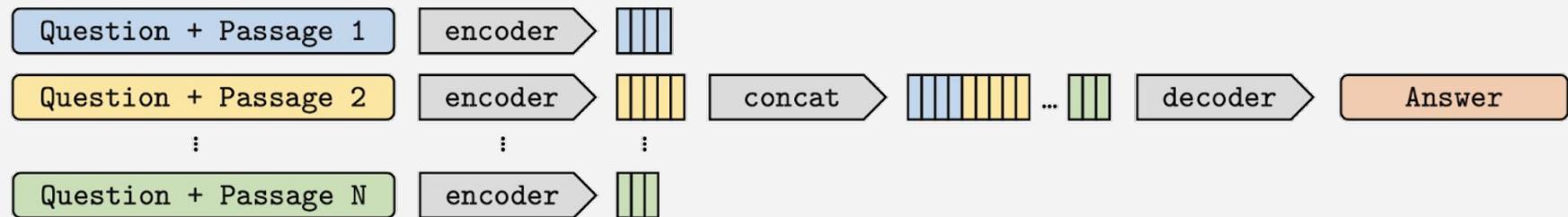
(Evidence)

OpenAI is launching a premium and paid-for version of ChatGPT. The free app will remain available. But it is liable to go offline during busy periods – and, during those, the people who have paid its monthly fee will have priority access. That is just one of the perks offered in return for the \$20 subscription to “ChatGPT Plus”.

ARA adopts the Extractive Question Answering model to improve accuracy

3. Architecture - Model

Fusion in Decoder

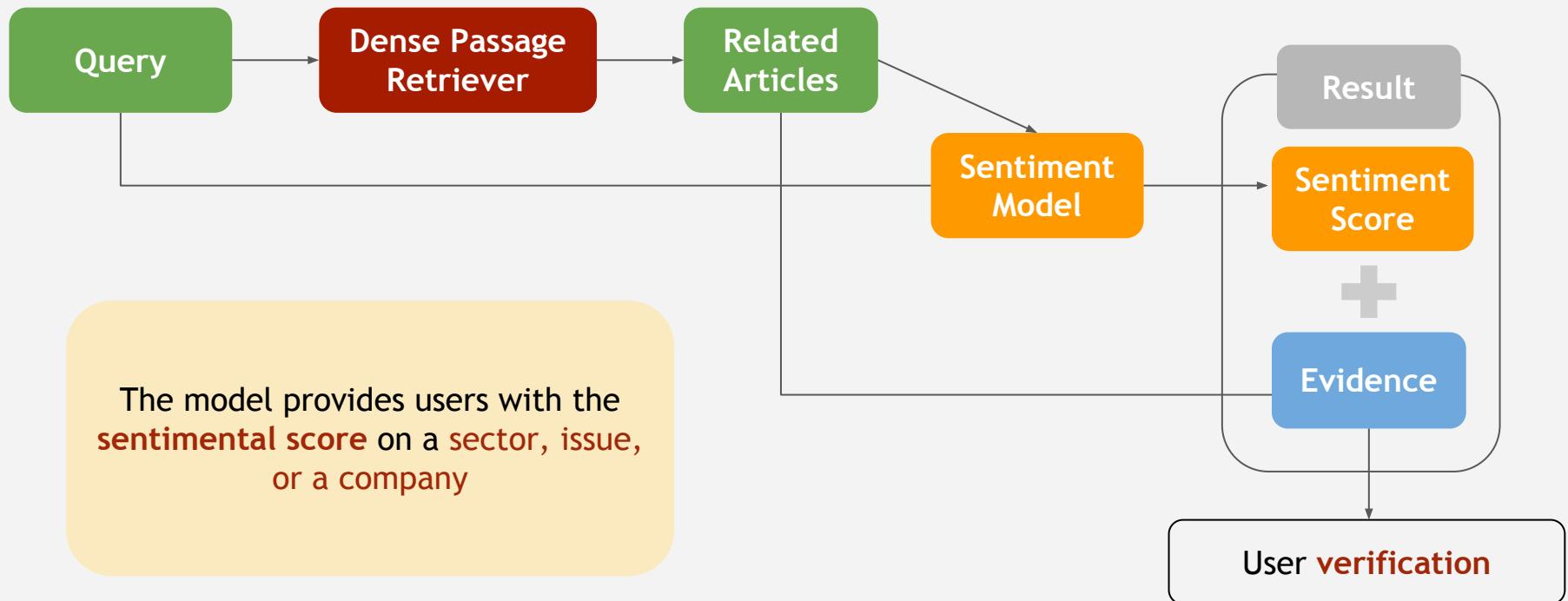


1. Our QA model use all N articles to generate answers to a given question
2. Error tracking is possible because user can see the generated answer and the rationale at the same time

3. Architecture - Model

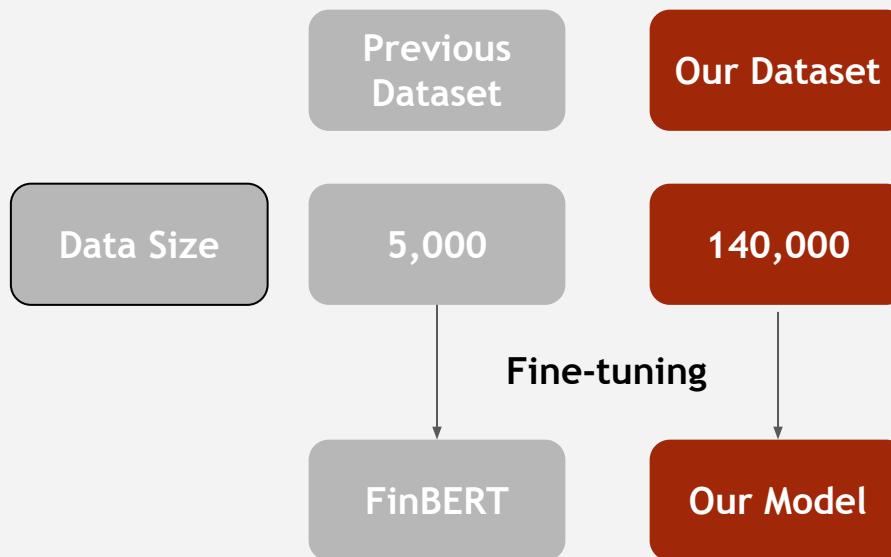


Sentiment Analysis



3. Architecture - Model

Sentiment Analysis



[Example 1]

Pre-tax loss totaled euro 0.3 million, compared to a loss of euro 2.2 million in the first quarter of 2005.

True Label: **Positive**

FinBERT: **Negative** Ours: **Positive**

[Example 2]

This implementation is very important to the operator, since int is about to launch its Fixed to Mobile convergence service in Brazil.

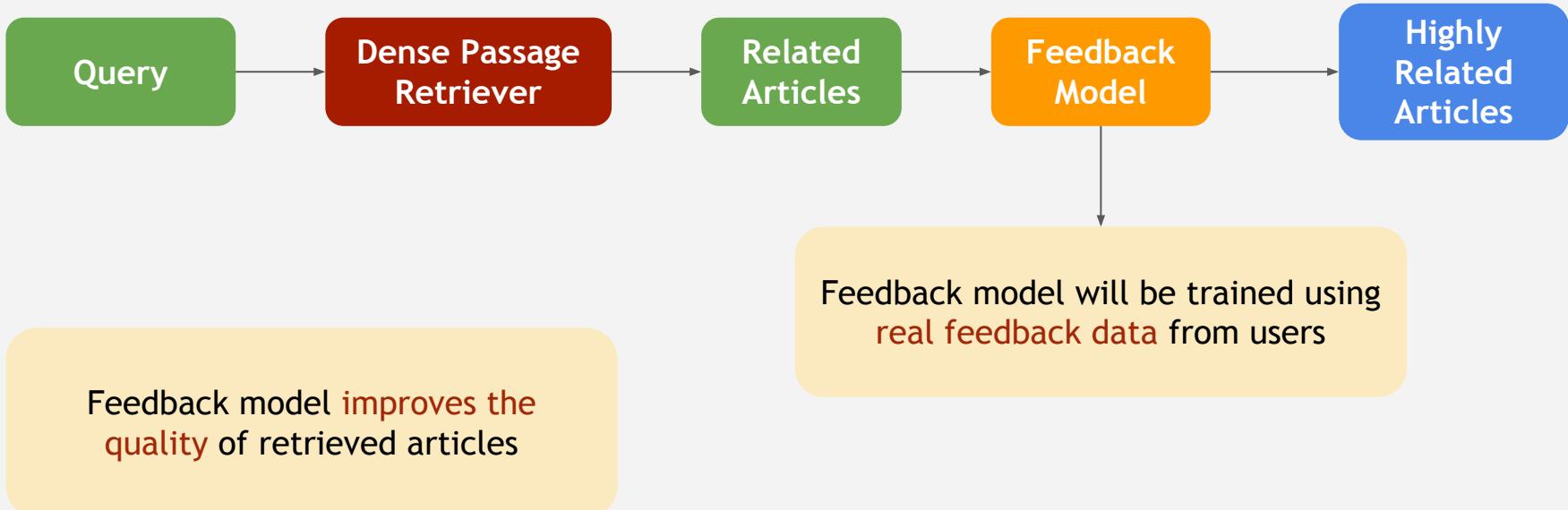
True Label: **Neutral**

FinBERT: **Positive** Ours: **Neutral**

3. Architecture - Model



Feedback Model



3. Architecture - Model

Sentence Transformer

```
train_dataloader = DataLoader(
    sts_train_examples,
    shuffle=True,
    batch_size=train_batch_size,
)

dev_evaluator = EmbeddingSimilarityEvaluator.from_input_examples(
    sts_valid_examples,
    name="sts-dev",
)

test_evaluator = EmbeddingSimilarityEvaluator.from_input_examples(
    sts_test_examples,
    name="sts-test",
)

[14] embedding_model = models.Transformer(
    model_name_or_path=retrained_model_name,
    max_seq_length=2048,
    do_lower_case=True
)

pooling_model = models.Pooling(
    embedding_model.get_word_embedding_dimension(),
    pooling_mode_mean_tokens=True,
    pooling_mode_cls_token=False,
    pooling_mode_max_tokens=False,
)

model = SentenceTransformer(modules=[embedding_model, pooling_model])

train_loss = losses.CosineSimilarityLoss(model=model)

warmup_steps = math.ceil(len(sts_train_examples) * sts_num_epochs / train_batch_size + 0.1)
logging.info(f'Warmup steps: {warmup_steps}')

model.fit(
    train_objectives=[(train_dataloader, train_loss)],
    evaluator=dev_evaluator,
    epochs=sts_num_epochs,
    evaluation_steps=int(len(train_dataloader)*0.1),
    warmup_steps=warmup_steps,
    output_path=sts_model_save_path
)
```

```
test_evaluator(model) # original model
0.474170203374172
test_evaluator(model, output_path=sts_model_save_path) # fine-tuned model
0.8345196765882963
```

```
MODEL_SAVE_REPO = PATH+'ARA_Sentenceformer'
HUGGINGFACE_AUTH_TOKEN = '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'

model.push_to_hub(
    MODEL_SAVE_REPO,
    use_temp_dir=False,
    use_auth_token=HUGGINGFACE_AUTH_TOKEN
)
```

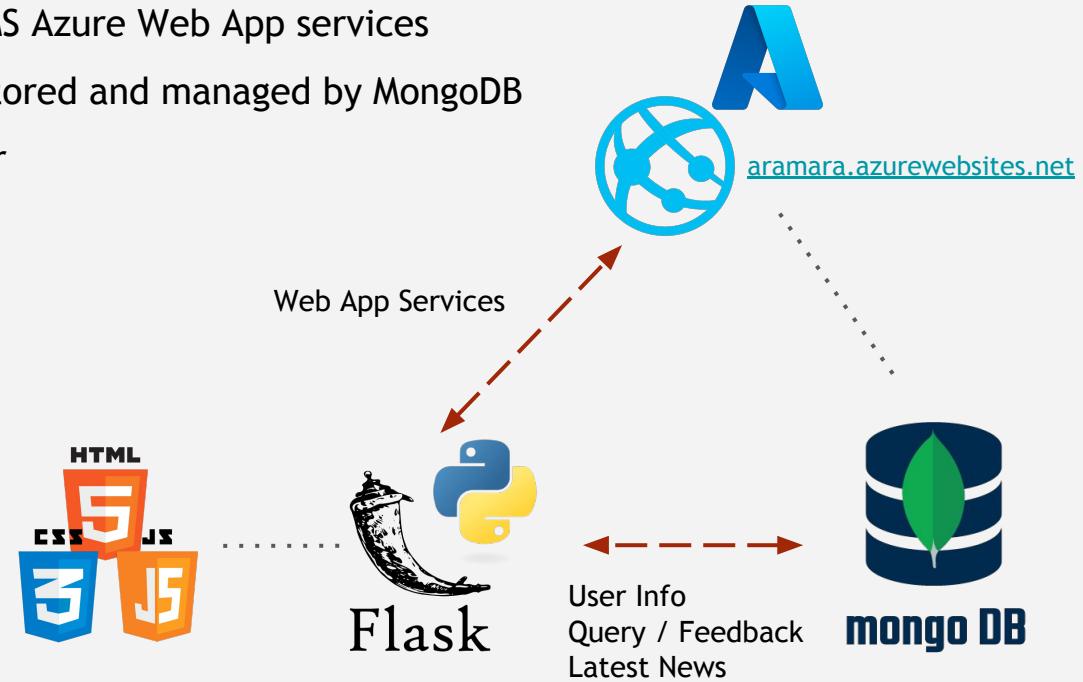
Feedback Model nearly doubles the accuracy!

3. Architecture - Web Application



Key features

- Flask Web Framework deployed on MS Azure Web App services
- User Info / Query / Feedback data stored and managed by MongoDB
- Latest News provided for each sector

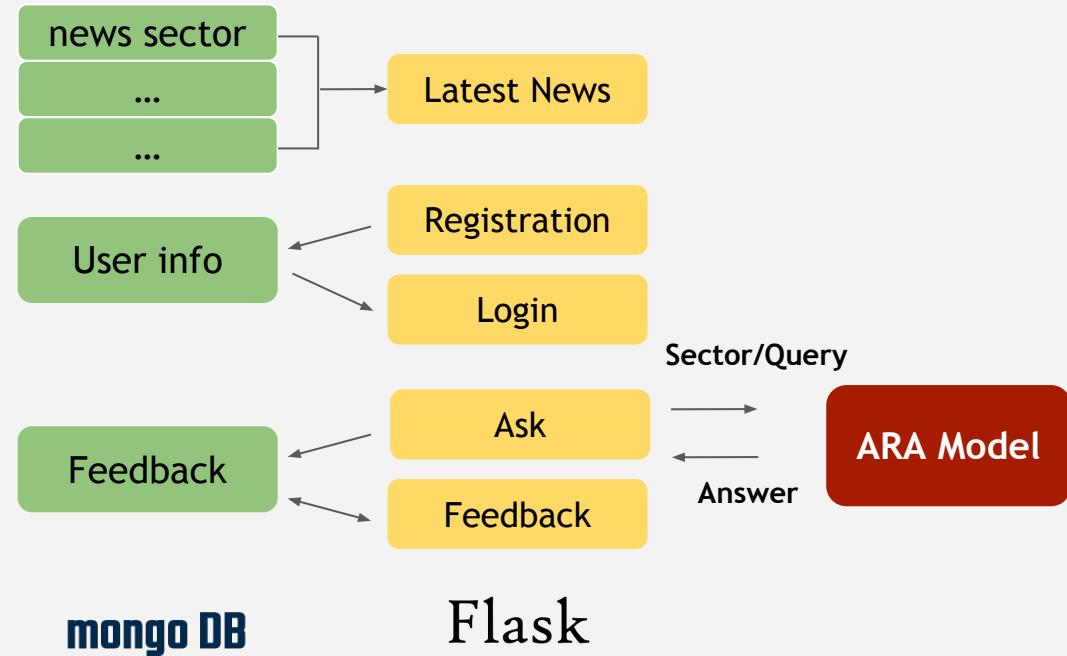


3. Architecture - Web Application



REST APIs

- Latest News
- User Registration & Login
- Ask Query & Feedback



3. Architecture - Web Application



User Interface

Welcome to ARA

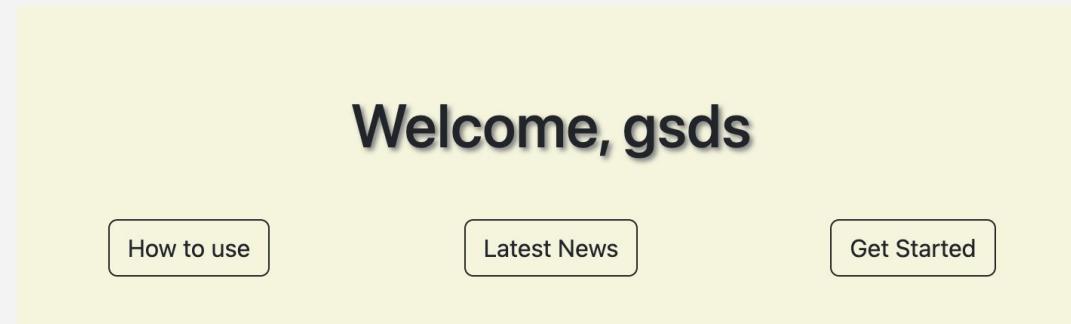
ID _____ Password _____

[Login]

Welcome to ARA

p4ds _____
.....
gsds _____
gsds _____
gsds@gsds.snu.kr _____

[Register]



[Welcome Page]

3. Architecture - Web Application



User Interface

Assistant for Research Analysts ≡

ARA is here to provide you with a comfortable morning

Please don't spend any more time struggling to find and read news articles every morning!

ARA can select and summarize recent updates in your sector for you

What ARA can do

- Reduce the time to select and read news articles related to your interests
- Answer your open domain question with reliable evidence
- Provide accurate and up-to-date financial data
- Provide sentiment analysis on recent articles related to your interests

[How to use]

Assistant for Research Analysts ≡

Latest news in ...

BASIC_MATERIALS

- Scoop Up These 4 Promising Interest Coverage Ratio Stocks
- Royal Gold (RGLD) Gives Update on Penasquito Operation Halt
- 10 Best Recession-Proof Stocks to Buy Now

COMMUNICATION_SERVICES

- Asian stocks rise on Fed pause bets, China lags on weak inflation
- A Waymo self-driving car hit and killed a 'small dog' near a San Francisco homeless encampment. The company sends 'sincere condolences' to the unknown owner
- TikTok-Owner Tests ChatGPT-Style Bot After Joining China AI Race

CONSUMER_CYCLICAL

- Thinking About Buying Nvidia Stock? These AI Stocks Might Be Better Options.

[Latest News]

Assistant for Research Analysts ≡

Basic Materials

what is 10 Best Recession-Proof Stocks to Buy Now

submit

e.g.give me a summary of latest news

Answer: Valvoline Inc., PPG Industries Inc., and Agilon Health Inc.

Evidence.

Top growth stocks this month include Valvoline Inc., PPG Industries Inc., and Agilon Health Inc., all of which more than doubled their earnings in the most recent quarter. Growth stocks have rebounded from their 2022 lows, driven higher by the buzz around artificial intelligence and anticipation of an end to Federal Reserve rate hikes. Below we look at the top growth stocks across three categories: fastest earnings-per-share (EPS) growth, fastest sales growth, and combined EPS and sales growth.

★★★★★ Continue Done

[Ask ARA]

3. Architecture - Web Application

REST APIs

```
@app.route('/latest')
def latest():
    latest_json=[]
    for sector in sectors:
        sector_db = db[sector]
        cursor = sector_db.find().sort("_id", -1).limit(3)
        latest=list(cursor)
        titles = []
        for i in range(len(latest)):
            news = latest[i]
            titles.append(news['title'])

        sector_json = {
            'sector' : sector,
            'titles' : titles
        }
        latest_json.append(sector_json)
    return dict(success = 1, result = latest_json)
```

[Latest News]

```
@app.route('/register/<newid>/<newpwd>/<newFirst>/<newLast>/<newEmail>', methods=['GET'])
def insertOne(newid, newpwd, newFirst, newLast, newEmail):
    queryObject = {
        'ID': newid,
        'PWD': newpwd,
        'Name' : {'First' : newFirst, 'Last' : newLast},
        'newEmail': newEmail
    }
    val = user_db.find_one({"ID" : newid})
    if val:
        return dict(success=0)
    else:
        query = user_db.insert_one(queryObject)
        return dict(success=1)

@app.route('/valid/<id>/<pwd>', methods=['GET'])
def validate(id, pwd):
    query = user_db.find_one({"ID" : id })
    if query:
        if query['PWD'] == pwd:
            return dict(success=1, username = query['Name']['First'])
        else:
            return dict(success=0, username = 1)
    else:
        return dict(success=0, username = 0)
```

[User Registration & Verification]

```
@app.route('/ask/<sector>/<query>', methods=['GET']) #sector query answer -
def ask(sector, query):
    if query:
        # connect to model output
        url = "https://47d1-34-173-102-48.ngrok-free.app/QA"

        response = requests.get(url, {'input_text': sector + ';' + query})
        answer = response.text

        feedback = {
            'sector' : sector,
            'query' : query,
            'answer' : answer,
            'rate' : None
        }
        p = feedback_db.insert_one(feedback)
        key = str(p.inserted_id)

        return dict(success=1, result=[query, answer, key])

#db에 key 찾아서 rate랑 update
@app.route('/feedback/<key>/<rate>', methods=['GET'])
def feedback(key, rate):
    feedback_db.update_one({'_id': key}, {"$set": {'rate' : rate}})
    return dict(success=1, result=rate)
```

[Ask Query & Feedback]

4. Project Demo



Welcome, gsds

How to use

Latest News

Get Started

5. Conclusion



Limitation

- The article database relies solely on Yahoo Finance as its source
- User should specify a sector before asking a question
- Limited timeframe available to collect feedback data

Future Work

- Build feedback database
- More user friendly UI/UX
- Upgrade article resources

How we spent our last 3 months...



Project for Data Science / Meeting Minutes



Share



Meeting Minutes

Calendar view

Filter

Sort



...

New

Name	Date	Tags
KICK OFF	March 9, 2023	
교수님 면담	March 14, 2023	
Project Proposal 준비	March 21, 2023	
Project Proposal 준비 2	April 4, 2023	309호
Meeting Minutes	April 11, 2023	309호
Meeting Minutes 4/20	April 20, 2023	309호
교수님 면담2 오후4:15	April 25, 2023	407호
Meeting Minutes	May 2, 2023	309호
Meeting Minutes 5/10	May 10, 2023	300호
Meeting Minutes 5/16	May 16, 2023	407호
5/23 Meeting	May 23, 2023	309호
교수님 면담3 오후4:15	May 30, 2023	309호
To-do 5/30	May 30, 2023	
Final Meeting Minutes 6/7	June 7, 2023	309호
Final Presentation	June 11, 2023	309호

minwl / ARA (Private)

Code Issues Pull requests Actions Projects Security Insights

main · 1 branch · 0 tags

Go to file Add file · < Code · About

No description, website, or topics provided.

Readme

Activity

1 star

1 watching

0 forks

Releases

No releases published Create a new release

Packages

No packages published Publish your first package

Contributors 5

README.md

A screenshot of a GitHub repository page for the user 'minwl' named 'ARA'. The repository is private. The main tab is selected, showing a list of commits. The commits include: 'minwl url update' (4b6681a, 5 days ago), '.github/workflows' (Update azure-functions-app-python.yml, 3 weeks ago), 'BackEnd' (Add files via upload, last week), 'KPMG_flask' (KPMG example, last month), 'crawling' (separate DB for each sector n daily schedule, 2 weeks ago), 'flask' (url update, 5 days ago), 'DS_Store' (update howto.html, 3 weeks ago), 'Colab_yahoo_sector.ipynb' (added url in dataset, 2 weeks ago), 'KPMG_FINAL.ipynb' (KPMG example, last month), 'P4DS_Colab_Server.ipynb' (Add files via upload, last month), 'Python-Postgresql_Connection.ipynb' (Add files via upload, last month), 'README.md' (Update README.md, last month), 'SentenceBert_Finetuning.ipynb' (SBERT_Finetuning, 3 weeks ago), and 'test.py' (modified html and js, 3 weeks ago). The repository has 5 contributors and no releases.



Thank you A yellow smiley face emoji with a small blue halo above its head.