웹 애플리케이션 서버(WAS)에서의 검색 증강 생성(RAG) 기술을 이용한 지식 기반 QA 문제 해결

전준형, *김상철, 김주철†, 윤성준†
국민대학교 소프트웨어 융합대학원, 조이시티†
jhjun1987@kookmin.ac.kr, *sckim7@kookmin.ac.kr, kchaos@joycity.com,
tjdwns4302@joycity.com

Knowledge-based QA problem solving in web application servers (WAS) using Retrieval-augmented generation (RAG) technology

Joon Hyoung Jun, Sang-Chul Kim* Joo Chul Kim, Seong Joon Yoon
Kookmin Univ*, Joycity†

요 약

본 논문은 OpenAI의 ChatGPT와 같은 사전 학습된 대규모 언어 모델을 실제 서비스 환경에 적용하는 방법을 탐구한다. 검색 증강 생성 기술을 활용하여 벡터 검색과 클러스터링 기법을 통해 자연어 검색 데이터베이스를 구축하고, 이를 서비스에 적용하는 과정을 상세히 설명한다. 또한, 이러한 기술을 전통적인 웹 응용 서버(WAS)와 통합하고, 지식기반 질문-답변(QA) 시스템을 통해 그 성능과 효과를 체계적으로 분석한다. 이 과정에서 OpenAI ChatGPT의 통합이 실제 서비스에 어떻게 기여할 수 있는지, 그리고 향후 어떤 발전 가능성이 있는지에 대해 심도 깊게 논의한다.

Ⅰ. 서 론

2022년 11월 chatGPT 3.5 Turbo의 공개 이후, 인공지능과 대규모 언어 모델(LLM)을 이용한 자연어 생성(NLG) 분야에 뜨거운 관심이 집중되고 있다. [1]

Figure 1. Google Trend

이는 구글 트렌드를 통해서도 확인이 가능하다. 2022년 11월 공개 이후, 구글 검색 횟수는 폭발적으로 증가하였다. [2]

그러나 LLM이 실제 사실과 다른 거짓된 답변을 하는 이른바 환각(Hallucination) 문제가 대두되었다. [3] 환각은 LLM 기반 NLG 결과물에 대한 신뢰성을 저해하여 해당 기술이 보급되는데 있어서 큰 걸림돌이었기 때문에 이를 해결하기 위해 다양한 기술적 시도를 하게 된다.

미세조정(Fine-Tuning)과 같이 대규모의 데이터셋을 준비하고 모델을 학습시켜 파라미터를 미세조정하는 방법과 함께, 이러한 그라디언트 업데이트 대신 입력 프롬프트에 몇 건의 예제(Few shots examples)를 제공함으로써 LLM 모델이 이를 기반하여 답변을 생성하는 방법이 제시되었다. [4]

Few-Shot (FS)과 유사한 방법은 In-Context Learning (ICL)과 Retrieval-Augmented Generation (RAG)이 있다 [5], [6]

FS, ICL, RAG 모두 모델의 그라디언트를 조정하지 않고, 입력 프롬프트에 참조할 만한 예제나 자료를 전달하여 답변 품질을 개선한다는 공통점이 있다. 다만 모든 분야에 걸쳐서 사전에 자료를 준비하는 것은 현실적으로 어려우므로, 사전에 준비된 특정 도메인의 자료에 기반하여 해당 도메인에 한정된 활용에 보다 적합성을 보이게 된다.

그러나 chatGPT를 비롯한 LLM 언어 모델의 경우, 입력 길이에 제한이 있다. 물론 최근에는 GPT-4-Turbo가 나오며 128k Token까지 확장되어 보다 광범위하게 ICL을 사용할 수 있으나, openAI의 가격정책은 입력 token 수에 비례하여 측정이 되므로, 모든 문서를 제공하는 것보다 자연어 검색을 통해 꼭 필요한 자료만 입력으로 추가하는 것이 보다 효율적이다. [7]

Model	Input (1K tokens)	Output (1K tokens)
gpt-4-turbo	$0.01	$0.03
gpt-4	$0.03	$0.06
gpt-4-32k	$0.06	$0.12
gpt-3.5-turbo	$0.001	$0.002

Figure 2. chatGPT pricing

본 논문에서는 현재 상용화되어 서비스되고 있는 openAI의 chatGPT를 기반으로 하여 특정 도메인에서 지식 기반한 QA문제를 해결할 수 있도록 하기 위해 자연어 검색이 가능한 DB를 구축하고 이를 활용하여 실제 서비스 가능하도록 WAS 적용하고 그 성능을 분석하였다.

Ⅱ. 본론

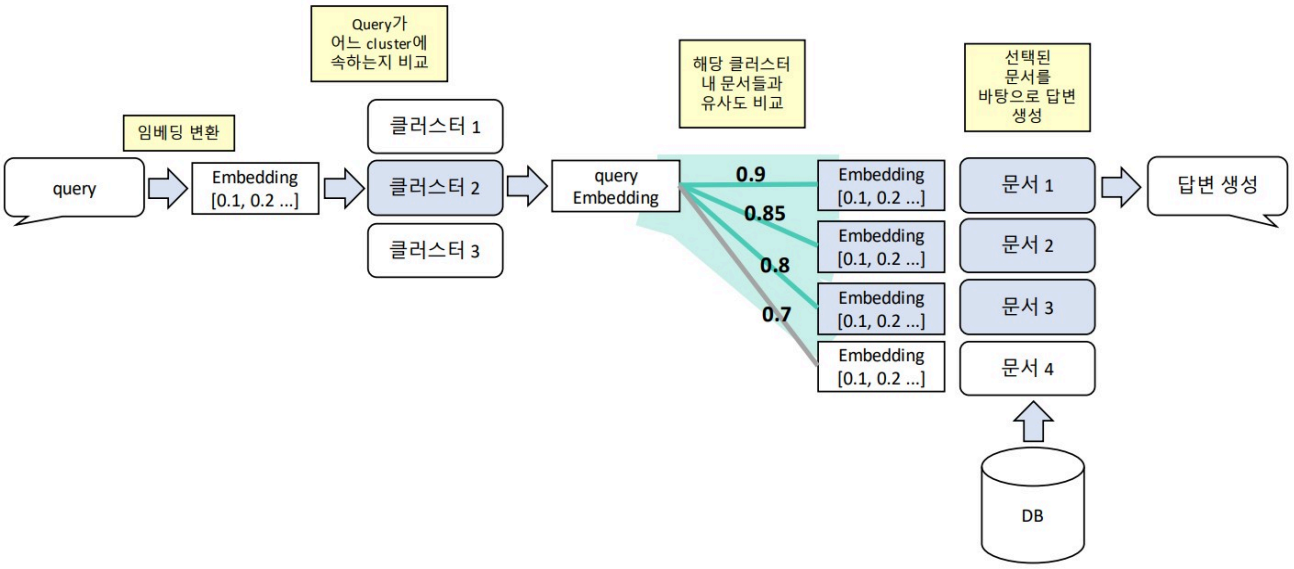

Figure 3. 기본적인 구성도

사용자의 질의(Query)에 관련된 문서를 검색하는 방법은 여러 가지가 있지만, 본 논문에서는 사용자 Query를 Embedding Model을 통해 Vectorize하여 이를 비교하여 cosine similarity 가 높은 문서를 찾는 Vector Search 기법을 이용하였다. [8]

자연어를 벡터로 만드는 모델 또한 여러가지가 있으나 본 논문에서는 openAI에서 제공하는 Embedding Model인 text-embedding-ada-002를 이용하여 Vectorize하였다.

MODEL NAME	MAX INPUT TOKENS	OUTPUT DIMENSIONS
text-embedding-ada-002	8191	1536

Figure 4. Embedding Model 정보

text-embedding-ada-002 모델을 사용하면 1536 차원의 벡터가 결과값으로 나오게 된다. 언어적으로 의미가 유사한 두 문장은 임베딩한 벡터 또한 그 거리와 방향이 가깝게 나온다.

$$cos\ similarity = \frac{\widehat{A} \cdot \widehat{B}}{||\widehat{A}|| \times ||\widehat{B}||}$$

이를 Java 로 구현하면 아래와 같다

```
public double cosineSimilarity
(double[] vectorA, double[] vectorB)
{
    double dotProduct = 0.0;
    double normA = 0.0;
    double normB = 0.0;

    for (int i = 0; i < vectorA.length; i++)
    {
        dotProduct += vectorA[i] * vectorB[i];
        normA += Math.pow(vectorA[i], 2);
        normB += Math.pow(vectorB[i], 2);
    }
    return dotProduct / (Math.sqrt(normA)
    * Math.sqrt(normB));
}
```

위의 cosineSimilarity 함수를 이용하여, 사용자의 Query와 사전에 입력된 자료들의 유사도를 계산하면, 사용자의 쿼리와 의미적으로 가까운 문서를 검색할 수 있다.

그러나 MySQL과 같은 전통적인 RDBMS에서는 벡터의 인덱싱이 불가능하다는 단점이 있다. 인덱싱을 하지 않는다면 사용자가 Query를 날릴 때마다 모든 문서를 색인해야하는 단점이 발생하게 된다.

이를 개선하기 위해 다양한 방법들이 제안되고 있으나, 여기서는 K-Means나 CLALANS, DBSCAN과 같은 클러스터링 알고리즘을 사용하여 군집을 미리 나누어 놓는다. 이후에 사용자가 Query를 입력하게 되면 앞서 사용했던 clustering 알고리즘을 사용하여 사용자의 쿼리가 속하는 군집을 찾게 된다. 이후 해당 군집 내에 속하는 문서들과 cosine similarity 점수를 구하여 유사한 문서를 찾게 된다.

이러한 clustering Algorithm을 제공하는 Java 라이브러리는 여러가지가 있지만 여기서는 Smile을 사용했다. [9]

```
// ArrayList<double[]>을 double[][]로 변환
double[][] data = new
double[dataList.size()][dataList.get(0).length];
data = dataList.toArray(data);

    // 문서의 수에 기반하여 클러스터링을 만들게 한다
int k = 1 +  dataList.size() / n ;

    // 각각의 클러스터링 모델을 학습시킨다
KMeans kmean = KMeans.fit(data, k);
CLARANS<double[]> clarans =  CLARANS.fit(data,
new EuclideanDistance(), k, k);
DBSCAN<double[]> dbscan = DBSCAN.fit(data, 1,
0.5);
DeterministicAnnealing da
=DeterministicAnnealing.fit(data, k);
```

이렇게 fitting된 clustering Model을 사용하면 아래와 같은 결과가 나온다. 이를 다시 db에 저장하게 되면, 아래 그림처럼 각각의 문서들에 대해서 인덱싱 형태로 활용이 가능하게 된다.

본 논문에서는 클러스터링 알고리즘별 성능 측정 및 비교에 대해 상세히 다루지 않으나, 일반적인 특징을 간략히 언급한다.

k-means는 클러스터 중심을 평균값으로 계산하여 군집화를 수행하는 방식으로, 비교적 간단하고 계산이 빠르다는 장점이 있다. 그러나 특정 군집 형태(비구형 등)에 대해서는 실제 군집 분포와 다르게 오분류가 발생할 수 있으며, 이는 군집의 크기나 밀도가 균일하지 않은 경우 더욱 두드러질 수 있다.

반면, DBSCAN은 밀도 기반 클러스터링으로, 데이터 포인트의 밀도를 기반으로 군집을 형성한다. 이 방식은 k-means보다 다양한 형태의 군집을 더 잘 찾을 수 있지만, 매개변수(이웃의 반경 및 최소 포인트 수) 설정에 민감하며, 의미적으로 고립된 데이터 포인트를 노이즈로 간주하여 군집에서 제외하는 경향이 있다. [10]

Clarans는 medoids를 기반으로 하는 분할 방식 클러스터링으로, 반복적으로 메도이드를 탐색하고 군집을 재분류한다. 이 과정에서 다양한 후보 메도이드 조합을 탐색하여 최적의 군집화를 찾는 것이 목표다. [11]

Deterministic Annealing은 높은 온도에서 시작하여 점진적으로 온도를 낮추며 최적화를 수행하는 알고리즘으로, 초기에는 넓은 탐색 공간에서 후보 해를 탐색하고 온도가 낮아짐에 따라 점점 더 정교한 탐색을 수행하여 최종적으로 지역 최소값을 찾아 군집을 분류한다. 이 과정은 시스템의 자유 에너지를 최소화하는 방향으로 진행된다. [12]

이러한 각기 다른 클러스터링 방식은 데이터의 특성 및 요구 사항에 따라 적절하게 선택되어야 하며, 각각의 방식이 가지는 장단점을 고려하는 것이 중요하다.

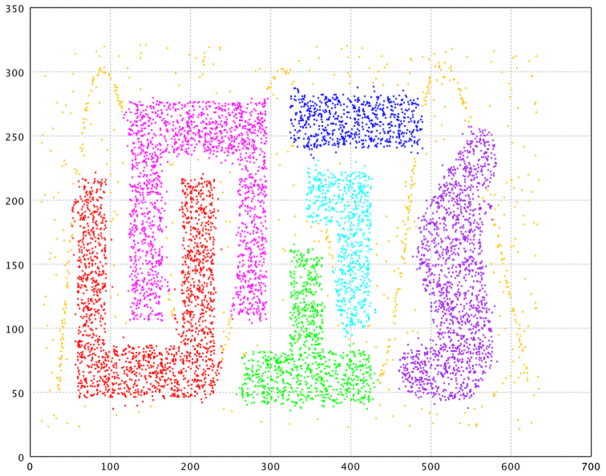

Figure 5 - DBSCAN과 노이즈 [13]

content	embedding_json	kmean	dbscan	clarans	da
라이선스 지급 : 지급이 필요한 라이선스 및 프...	[0.009119162, -0.019891467, 0.00000629024...	9	214748...	5	5
무선 Wifi 안내 서현 퍼스트타워 무선 Wifi 안내 ...	[0.005591829, -0.007846229, 0.01632068, -0....	6	2	1	5
공인(public) IP 안내 서현 퍼스트타워 공인IP 안...	[0.0034168193, -0.018408986, 0.018090215, ...	6	2	1	5
사내 계정 보안 정책 사내 패스워드 정책 1) 암...	[-0.009138715, 0.0093713375, 0.012368836, ...	8	1	8	1
사내 계정 패스워드 변경 방법 패스워드 변경 방...	[0.00056407484, 0.00005884234, -0.0103952...	8	1	8	1
사내 PC 원격접속 안내(서현) : 외부에서 서현 ...	[-0.011087109, -0.036170714, 0.02571266, -0...	6	0	1	5
사내 PC 원격접속 안내(수내) : 외부에서 수내 ...	[-0.011191284, -0.039087735, 0.027157342, -...	6	0	1	5
정보보호파트 업무-정보반출 정보반출 1) 개인...	[-0.006336592, 0.0037354294, 0.0106308395,...	4	3	2	9
정보보호파트 업무-계약서 계약서 1) 개인정보 ...	[-0.0049012583, -0.014277293, -0.000484712...	4	3	2	9
정보보호파트 업무-보안성 검토 보안성 검토 1)...	[0.016620787, -0.016819602, 0.009112274, -...	4	3	2	9
정보보호파트 업무-침해사고 침해사고 1) 해킹...	[-0.0023179017, -0.023821939, 0.014793965,...	4	214748...	1	9
정보보호파트 업무-DLP DLP 1) 반출 신청 방법 ...	[-0.010115241, -0.0059396853, 0.0002286913...	4	3	2	9
정보보호파트 업무-VMFORT VMFORT 1) 설치 ...	[-0.012413951, -0.01692934, 0.0055903215, -...	4	4	3	9
인프라 구축 조이시티에서 사용중인 클라우드 ...	[0.0035135292, -0.026136423, 0.010964064, ...	2	0	7	7
인프라 운영(클라우드 Console IAM 계정 생성) ...	[-0.007730825, -0.025605025, 0.0071176905,...	2	0	7	7
AWS 계정 생성 및 권한(IAM) 신청서 안내 신청 ...	[-0.011272703, -0.02450474, 0.0032410654, -...	2	0	7	7
GCP 프로젝트 생성 및 권한(IAM) 신청서 안내 ...	[-0.00464769, -0.031984102, 0.0055805594, -...	2	0	7	7
무료 와이파이가 필요해!	[-0.014356962, -0.02807287, 0.0050917133, -...	3	5	6	6
무료 인터넷을 알려줘!	[0.0020354271, -0.012941377, 0.027497964, ...	3	5	6	2
방문객들을 위한 인터넷이 필요해!	[-0.006619625, -0.0024083392, 0.032737534,...	3	5	6	2
방문객들을 위한 와이파이를 알려줘!	[0.0061711883, -0.0097037535, 0.023661206,...	3	5	6	6
사내에서 쓸 수 있는 무선 인터넷 와이파이를 알...	[0.0019532293, -0.0072028404, 0.02907376, ...	6	5	6	6
모니터 지급이 필요합니다.	[-0.01303703, -0.020417225, 0.0014224306, -...	3	214748...	6	0

Figure 6. embedding vector와 clustering

사용자의 Query도 fitting된 모델을 통해 Clustering을 하게 되면 해당되는 군집의 index 번호가 나오게 된다. 사용자의 쿼리(Query)가 속한 군집에 포함된 문서들과 Cosine Similarity를 비교하여 Top 3 문서를 선택한다.

이제 검색된 문서를 Prompt에 포함하여 LLM에 입력 메시지로 전달하게 된다.

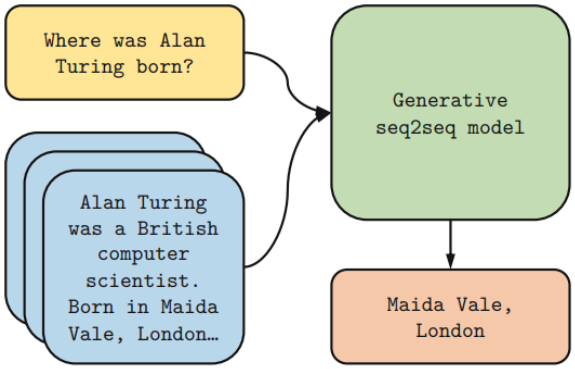

Figure 7. Fusion-in-Decoder 개념 [14]

Fusion-in-Decoder의 경우, 사용자 Question과 연관된 Passages 문서들을 묶어서 encoding한 후, 이를 decoder에 전달하기 전에 하나로 합쳐서 보내게 된다. [14]

그러나 openAI와 같은 상용 서비스 LLM의 경우, Encoder와 Decoder가 따로 분리되어있지 않아 Fusion-in-decoder와 같은 방법을 사용할 수 없다. 따라서 In-context Learning이나 Few-Shot Learning에서 했던 것처럼 검색된 문서들을 Query와 연결하여 하나로 합친 다음 이를 Encoder에 보내는 형태를 취하였다.

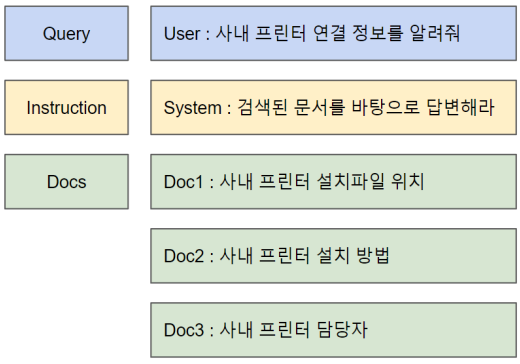

Figure 8. Query + Docs

병합의 위치를 Fusion-in-Decoder와 달리 변경하여 Encoder 앞에 위치시켰을 때 정확도나 답변의 품질에 영향이 있는지 비교하여 확인하지 못한 것은 아쉬움이 남는다. 이는 후속 연구를 통해 보다 심도 있게 다룰 필요성이 있다.

Figure 9. 개발된 화면

실험 결과

USMLE 문제를 가지고 chatGPT를 성능 평가한 연구가 있다. [15] 이런 연구에서 아이디어를 얻어 한국 공인중개사 시험 문제를 기반으로 성능 평가를 하였다.

chatGPT-4의 장점은 별도의 예제 제공이 없이 zero-shot 상태에서 성능 테스트를 해도 점수가 월등하게 높은 점이다. 지나치게 고득점이 나오는 분야의 QA 테스트에서는 검색DB를 통해 RAG를 구현했기 때문에 점수가 잘 나오는 것인지, 아니면 chatGPT 4 자체가 학습이 잘 된 상태여서 점수가 잘 나오는 것인지 모호하다.

공인중개사 시험의 경우 각 과목별로 성격이 비교적 명확하여 클러스터링이 원만하게 된다는 장점이 있고, 또 부동산법은 각국마다 상이하여 Zero-shot 상태에서는 한국 공인중개사 제도에 맞지 않는 답변을 할 때가 있다는 것이다. 따라서 Zero shot과 비교하여 향상된 점수는 검색 증강 생성의 결과물이라 할 수 있다.

과목명	GPT-4 (RAG)	GPT-4 (Zeroshot)
부동산의 개념	7 / 7 (100%)	6 / 7 (85%)
부동산의 속성	2 / 2 (100%)	1 / 2 (50%)
부동산의 종류	8 / 10 (100%)	6 / 10 (60%)
부동산의 특성	4 / 5 (80%)	4 / 5 (80%)
합계	21/24 (87.5%)	17 / 24 (70%)

Figure 10. 공인중개사 시험 점수

Ⅲ. 결론

본 연구에서는 실제 공인중개사 시험문제를 활용하여 성능 테스트를 수행하였고, 그 결과 약 17.5%의 성능 향상을 목격하였다. 이러한 결과는 검색 기반의 자연어 생성 기술이 실질적인 응용 분야에서 효과적임을 입증하는 중요한 사례로, 해당 기술의 발전 가능성을 다시 한번 확인시켜 준다.

Spring Boot, MyBatis, Smile 및 chatGPT API를 통합한 구성을 사용하여 자연어 검색 결과를 기반으로 답변을 생성하는 서비스를 개발하고 구축하였다. 이 시스템은 자연어 처리와 데이터베이스 관리 기술의 결합을 통해 보다 정확하고 신속한 정보 검색 및 처리 능력을 제공한다. 현재 이 서비스는 실제 기업 환경에서 내부적으로 점진적으로 도입되고 있으며, 그 사용성과 효율성에 대한 긍정적인 피드백을 받고 있다.

본 연구에서 구축된 현재의 텍스트 기반 검색 데이터베이스는 초기 단계에 불과하다. 향후 연구에서는 이미지나 파일을 벡터화하여 자연어 검색에 통합하는 방법을 모색할 것이다. 이를 통해 더욱 다양하고 복잡한 데이터 유형을 처리할 수 있는 능력을 개발함으로써 서비스의 범위와 깊이를 대폭 확장할 계획이다.

이와 더불어, 현재 자연어 검색에만 초점을 맞추고 있는 검색 시스템을 발전시켜 전통적인 키워드 검색과의 통합을 모색하고 있다. 이러한 하이브리드 검색 시스템은 자연어 검색의 직관성과 키워드 검색의 정확성을 결합하여 사용자 경험을 향상시키고, 검색 품질을 더욱 높일 것으로 기대된다.

현재 구축한 검색 데이터베이스를 지속적으로 발전시켜나가는 한편, LLAMA 2와 같은 온프레미스 LLM 모델과의 통합을 모색하여 기술적 진보를 도모할 계획이다. 이러한 통합은 자연어 처리 능력을 더욱 강화하고, 보다 정교하고 다양한 데이터 소스를 활용할 수 있는 발판을 마련할 것이다. 이번 연구와 개발은 이 분야에서의 지속적인 혁신과 발전을 위한 중요한 기반을 제공할 것으로 기대된다.

참 고 문 헌

[1] Introducing ChatGPT (https://openai.com/blog/chatgpt)

[2] Google Trend - LLM, GPT, openAI (https://trends.google.co.kr/trends/explore?q=LLM,GPT,openAI&date=2022-09-01%202023-12-02#TIMESERIES)

[3] ZIWEI JI, "Survey of Hallucination in Natural Language Generation", 2022, (https://arxiv.org/abs/2202.03629)

[4] Tom B. Brown, "Language Models are Few-Shot Learners", 2020, (https://arxiv.org/abs/2202.03629)

[5] Qingxiu Dong, "A Survey on In-context Learning", 2023, (https://arxiv.org/abs/2301.00234)

[6] Patrick Lewis, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks", 2020, (https://arxiv.org/abs/2005.11401)

[7] openAI - chatGPT pricing (https://openai.com/pricing)

[8] Isa M. Apallius de Vos, "Comparing in context: Improving cosine similarity measures with a metric tensor", 2022, (https://arxiv.org/abs/2203.14996)

[9] Smile - Statistical Machine Intelligence and Learning Engine (https://haifengl.github.io/clustering.html#clarans)

[10] Martin Ester, "A density-based algorithm for discovering clusters in large spatial databases with noise", 1996, (https://cdn.aaai.org/KDD/1996/KDD96-037.pdf)

[11] R.T. Ng, "CLARANS: a method for clustering objects for spatial data mining", 2002, (https://ieeexplore.ieee.org/document/1033770)

[12] Christos Mavridis, "Online Deterministic Annealing for Classification and Clustering", 2021, (https://arxiv.org/abs/2102.05836)

[13] Smile - DBSCAN (https://haifengl.github.io/clustering.html#deterministic-annealing)

[14] Gautier Izacard, "Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering", 2020 (https://arxiv.org/abs/2007.01282)

[15] Prabin Sharma, "Performance of ChatGPT on USMLE: Unlocking the Potential of Large Language Models for AI-Assisted Medical Education" 2023, (https://arxiv.org/abs/2307.00112).

[16] Kurt Shuster, "Retrieval Augmentation Reduces Hallucination in Conversation" 2021, (https://arxiv.org/abs/2104.07567).