

## EkoBot: Türkçe Destekli Akıllı Sanal Akademik Danışman

*EkoBot: Intelligent Virtual Academic Advisor with Turkish Support*

Ayça KUMLUCA TOPALLI <sup>1</sup> 

<sup>1</sup>İzmir Ekonomi Üniversitesi, Mühendislik Fakültesi, Elektrik ve Elektronik Müh. Bölümü, İzmir, Türkiye

### Öz

Bu çalışmada üniversite öğrencilerine sanal danışman olarak yardımcı olabilecek, Türkçe destekli bir akıllı yazılım, EkoBot sunulmuştur. Bu yazılımla öğrencilerin sorularına doğru ve hızlı bir şekilde yanıt almaları hedeflenmiştir. Bunun için yapay zekâ destekli büyük dil modellerinden yararlanılmıştır. Büyük dil modelinin, eğitimi sırasında kullanılmamış, üniversite yönetmeliklerine dayalı yanıtlar verebilmesi için bu belgeler modele dışardan verilmiş ve “almayla artırılmış üretim” yöntemi kullanılmıştır. Önerilen sistemin performansını ölçmek için ucu açık ya da olumlu ve olumsuz yanıtlara sahip 100 adet soru üretilmiştir. Alma kısmında, soruya en çok benzeyen beş bağlam metni ile %100 başarımla elde edilmiştir. Üretim kısmında, yanıt ile bağlam benzerlikleri 0,82 olarak bulunmuştur. Ayrıca önerilen çözümün bir Web sayfası olarak çalışan bir prototipi hazırlanmış ve öğrencilerin kullanımına sunulmuştur.

**Anahtar Kelimeler:** Doğal dil işleme, kosinüs benzerliği, istem mühendisliği, büyük dil modelleri, yapay zekâ, almayla artırılmış üretim

### Abstract

In this study, EkoBot, an AI-powered software with Turkish language support, designed to assist university students as a virtual advisor, is presented. The goal of this software is to provide students with accurate and prompt answers to their questions. For this purpose, large language models supported by artificial intelligence were utilized. To enable the large language model to give responses based on university regulations, which were not included in its training, these documents were provided to the model externally, and a “retrieval-augmented generation” method was employed. To evaluate the performance of the proposed system, 100 questions with open-ended or positive and negative responses were generated. In the retrieval phase, 100% success was achieved with the five context texts most similar to the question. In the generation phase, the similarity between the responses and the context was found to be 0.82. Additionally, a prototype of the proposed solution was developed as a Web page and made available for the students.

**Keywords:** Natural language processing, cosine similarity, prompt engineering, large language models, artificial intelligence, retrieval augmented generation

## 1. GİRİŞ

Üniversiteler gibi yüksek eğitim kurumlarında, öğrencilerin dersler, kayıtlar, sınavlar gibi çeşitli konularda soruları olmaktadır. Bu soruların yanıtları, çoğunlukla eğitim ve öğretimle ilgili yönetmeliklerde olduğu halde, öğrencilerin bu yönetmeliklerden haberdar olmaması ya da sayfalarca uzunlukta belgeler içinde aradıkları yanıtları bulamamaları nedeniyle, sorularını akademik danışmanlara ya da idari çalışanlara iletmektedirler. Özellikle kayıt zamanı gibi yoğun dönemlerde, bu tür soruları tek tek yanıtlamak hem akademik hem de idari çalışanlar için zor bir durumdur. Ayrıca çalışanlar yanıtları yalnızca mesai saatleri içinde verebilirler. Öğrencilerin bir çalışana gitmeden önce, sorusunu zamandan bağımsız olarak yönelebileceği otomatik bir sistem oldukça verimli olacaktır.

Dijitalleşme sayesinde, kurumlarda üretilen ve saklanan verilerin büyük boyutlara ulaşması, verimliliği artırmakla birlikte, bu büyük verinin içinde istenilen bilginin bulunmasının zorlaşmasına yol açmıştır. Eğitim kurumlarında da çeşitli formatlarda (pdf, docx, html, vb.) ve dâhili veri tabanlarında saklanan ve gün geçtikçe çoğalan metinler içinde geleneksel yöntemlerle arama yapmak artık uygunluğunu yitirmektedir. Söz konusu otomatik sistemin bu metinleri ortak bir biçime sokup akıllı bir yöntem desteği ile hızlı ve doğru arama yapabilmesi gerekir.

Akıllı sohbet botları, yapay zekâ ve doğal dil işleme (Natural Language Processing, NLP) yöntemlerini kullanarak kullanıcıların sorularını anlayıp büyük dil modelleri (Large Language Models, LLM) sayesinde dilbilgisi kurallarına uygun yanıtlar üreten otomatik sistemlerdir [1].

LLM'ler, soru yanıtlama, tümce tamamlama, çeviri yapma gibi çeşitli metinler oluşturma amacıyla geliştirilen, büyük veri ile eğitilmiş milyarlarca parametrelerden oluşan, üretken yapay zekâ (Generative AI, GenAI) ürünleridir ve son zamanlarda hemen her alanda çok başarılı sonuçlar verdiklerinden kullanımları artmıştır. LLM temelli bir akıllı bot, eğitim kurumlarında öğrencilerin yönetmeliklerle ilgili sorularını yanıtlama biçiminde kullanılmaya uygun bir çözümdür.

Ancak bilindiği gibi bir LLM, ancak kendi eğitiminde kullanılan bilgilerle sınırlı ölçülerde doğru yanıtlar üretebilir. Kurumsal yönetmelikler gibi özel bilgiler, genel bir LLM eğitiminde kullanılmadığı için o kurumu ilgilendiren özel sorulara yanlış yanıt vermesi doğal bir sonuçtur. Öte yandan kullanıcıların bu teknolojiye duydukları güvenin sarsılmaması için, LLM'in sorulan sorulara doğru yanıt vermesi son derece önemlidir. Dolayısıyla, eğitim dışında kalan özel bilgilerle ilgili soru sorulan durumlarda LLM kullanmak isteniyorsa, bu bilgilerin de sisteme katıldığı ve soruların bu bilgilere dayanarak yanıtladığı bir çözüm kullanmak gerekir.

Almayla artırılmış üretim (Retrieval Augmented Generation, RAG.), LLM'in tekrar eğitilmesine ya da ince ayara tutulmasına gerek kalmadan, bir kuruma, konuya vb. özgü güncel bilgileri ve ek verileri kullanarak yanıtlar ürettiği yöntemdir [2, 3, 4]. LLM'lerin kullanım alanını genişlettiği, daha iyi yanıtlar verdiği ve yüksek bilgisayar gücü gerektirmediği için RAG yöntemi akademik ve endüstriyel araştırmacılarca benimsenmiştir [5, 6, 7].

RAG'in kullandığı ek veriler, sürekli olarak yenilenebilir ya da genişletilebilir ve bu durum LLM yapısını değiştirmez. Bu da büyük bir maliyet gerektirmediğinden RAG yönteminin bir avantajıdır. Ayrıca RAG sistemlerinde, sorulan sorunun yanıtı, verilen bağlamın içerisinde yoksa, LLM'in yanlış yanıt üretmek yerine soruyu yanıtlamak için yeterli bilgisi olmadığını söylemesi sağlanabildiğinden, güvenilirlerdir.

Bu çalışmada, RAG yöntemi kullanılarak, sorulara İzmir Ekonomi Üniversitesi "Ön Lisans ve Lisans Eğitim-Öğretim ve Sınav Yönetmeliği" [8] belgesinde

yer alan Türkçe bilgilere dayalı yanıtlar üreten, sorunun yanıtı bu belgede yoksa "bu soruyu yanıtlamak için yeterli bilğim yok" şeklinde dönüş yapan sanal bot, EkoBot, önerilmektedir. Bu sayede öğrenciler sorularına 24/7 yanıt alabileceklerdir. Sözü geçen yönetmelik gerçek hayatta kullanılan bir belgedir, dolayısıyla gerçekte karşılaşılabilecek durumlara uygun bir çözüm sunulmaktadır. Ayrıca, Türkçe destekli sanal akademik danışmanlık çözümü üzerine bir çalışmaya rastlanmamıştır. Bu açıdan burada önerilen sistem, bu alanda bir ilktir.

### 1.1. Literatür Taraması

Akıllı sanal sohbet botları birçok alanda kullanılmaktadır. Eğitim alanındaki botlarla ilgili akademik çalışmalardan en yenileri kronolojik sırada aşağıda özetlenmiştir.

RAG yöntemi geliştirilmeden, 2020 yılı öncesinde yapılan çalışmalarda, öğrencilerle sohbet botu altyapısı olarak yapay zekâ işaretleme dilinin (Artificial Intelligence Markup Language, AIML) sıklıkla tercih edildiği görülmektedir [9, 10, 11]. AIML, özellikle sohbet botları geliştirmek için kullanılan bir programlama dili olmuştur fakat yalnızca belli kalıplara uyan girdilere çıktı üretebildiği ve kural tabanlı olduğu için LLM yöntemine kıyasla kısıtlı başarımlar göstermiştir.

Uzun-kısa dönemli bellek (Long-Short Term Memory, LSTM), adlandırılmış varlık tanıma (Named Entity Recognition, NER) yöntemleri de bu dönemde botlarda kullanılan yapay zekâ öğrenme modellerinden olmuştur [12, 13]. Bu yöntemler, öz dikkat mekanizması fikrine dayalı olan dönüştürücü mimarisi üzerine kurulu LLM yöntemi kadar başarılı olamamıştır. Akıllı bot altyapısı olarak Amazon LEX'i [14], IBM Watson'u [15], Telegram'ı [16], Facebook Messenger'i [17] ya da DialogFlow'u [18] kullanan çalışmalar bulunmaktadır.

2020 yılı sonrasında, RAG yöntemini kullanmayıp diğer makine öğrenme teknikleriyle NLP yöntemlerini birleştiren akıllı sanal bot araştırmalarına rastlanmaktadır. Bunlardan Nguyen vd. [19] Rasa platformunu, Ula vd. [20] K-en yakın komşu algoritmasını, Verma vd. [21] RAG yönteminin bir parçası olan benzerlik kıstasını kullanmıştır.

RAG yöntemi, yukarıda sözü edilen geleneksel yapay zekâ algoritmalarına kıyasla daha doğru ve bağlam içi yanıt verdiği, modelin eğitiminde kullanılmayan verileri de kullanarak daha iyi genelleme yaptığı için tercih edilir duruma gelmiştir. Ayrıca RAG yapısında yer alan, dönüştürücü tabanlı LLM sayesinde, özel

sözcük kalıplarının kullanımını gerektirmeyen, daha doğal bir konuşma ortamı sağlanmıştır.

Üniversite ortamı için RAG destekli sohbet botu üzerine yapılan akademik çalışmalardan birinde, açık kod uygulama altyapısı olan Streamlit tercih edilmiştir [22]. Bu İngilizce bot çalışmasında vektör gömme işlemi Hugging Face platformundan “all-MiniLM-L6-v2” algoritması ile, LLM kısmı ise OpenAI GPT-3 modeli ile yapılmıştır. Başka bir çalışmada Maryamah vd. üniversite öğrenci adaylarının İngilizce sorularını yanıtlamak üzere RAG tabanlı bir bot tasarlamışlar ve vektör gömme işlemi için OpenAI şirketinin “Ada” algoritmasını kullanmışlardır. LLM olarak GPT-3.5 Turbo modelini seçmişlerdir [23].

Büyük dil modelleri zamanla gelişmektedir. Buna paralel olarak yapılan akademik çalışmalarda gün geçtikçe daha iyi modeller kullanılmakta ve daha iyi sonuçlar alınmaktadır; alınmaya da devam edilecektir. Bu konuda 2024 yılında yapılmış en güncel çalışmalardan birinde Neupane vd., OpenAI şirketinin şu ana kadar en iyi sonuç veren “text-embedding-3-large” algoritması ile vektör gömme işlemini yapmış, LLM olarak yine GPT-3.5 Turbo modelini kullanmışlardır [24]. Bu çalışma da diğerleri gibi İngilizce desteklidir.

İngilizce, yapısı gereği eksiz bir dil olduğu için gramer kurallarını yapay zekâ modellerine öğretmek daha kolaydır. Bu nedenle literatürdeki akıllı bot çalışmalarının büyük bölümü İngilizce için yapılmıştır. Eğitim alanında diğer diller için yapılmış sınırlı sayıda bot araştırmalarına örnek olarak şu çalışmalar gösterilebilir: Portekizce için Krassmann vd. tarafından geliştirilen e-öğrenme uygulaması [10], Endonezya dili için Chandra ve Suyanto’nun geliştirdiği NER uygulaması [12], Ukraynaca için Priadko vd. tarafından Telegram altyapısı ile geliştirilen model [16], Vietnamca için Nguyen vd. tarafından Rasa platformunda çalıştırılan uygulama [19] ve Nguyen vd. tarafından geliştirilen e-öğrenme modeli [25], Fransızca için Ait Baha vd. tarafından geliştirilen e-öğrenme uygulaması [26], Arapça için Aloqayli ve Abdelhafez’in geliştirdiği uygulama [27] ve Alharethi’nin geliştirdiği e-öğrenme modeli [28], Tayvanca için Lee vd. tarafından geliştirilen e-öğrenme uygulaması [29], İspanyolca için Martinez-Araneda vd. tarafından geliştirilen e-öğrenme uygulaması [30].

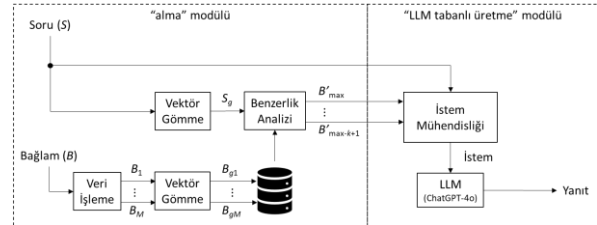
Bu bölümde sözü edilen ve kısaca tanıtılan çalışmaların birbirleriyle ve burada önerilen yöntem ile kıyaslaması için Tablo 1 hazırlanmıştır. Bu tabloda özellikle dil belirtilmeyen çalışmaların İngilizce destekli bir bot üzerine olduğu varsayılmıştır.

Yukarıda söz edilen çalışmalardan da görüldüğü gibi, Türkçe dili için gerçekleştirilmiş, eğitim alanında hizmet veren, RAG tabanlı akıllı bir sanal danışman botuna literatürde rastlanmamıştır. Ayrıca, LLM’lerin günümüzde en gelişmiş olan modeli GPT-4o ile yapılmış, Türkçe destekli, üniversite öğrencilerinin sorularını yanıtlamaya yönelik, RAG esaslı bir çalışma yoktur. Bu açılardan, bu çalışmada önerilen özgün sistem, bu alana bir yenilik getirmektedir.

## II. MATERYAL VE YÖNTEM

RAG, “alma” ve “LLM tabanlı üretme” modüllerinden oluşur [31]. LLM’e bir API üzerinden erişildiği ve uzaktan hizmet alındığı için güçlü bir yerel işlemciye gerek duyulmaz. Fakat, sorulan sorunun yanıtını bulmak için kullanılacak kurum içi bilgilerin de LLM’e bağlam olarak gönderilmesi gerekir. Bu durum iki temel sorun yaratır: ilk olarak LLM’lerin girdi olarak kabul ettiği istemlerin boyutu kısıtlıdır ve kurum belgeleri genelde bu boyutu aşacak uzunluktadır ve ikinci olarak, istemlerin boyutu arttıkça LLM’lerin ücretlendirmesi artar. Bu nedenlerden dolayı, kurum için belgelerden, sorulan soruyla en ilgili olanları bağlam olarak seçilmelidir. Bu seçme işlemini “alma” modülü gerçekleştirir. “LLM tabanlı üretme” modülü ise, soru ve seçilen kurum bilgilerini bir istem haline getirip LLM’e verir, LLM’in NLP kuralları ile ürettiği yanıtı kullanıcıya sunar.

Önerilen sistemin blok şeması Şekil 1’de verilmiştir.



Şekil 1. RAG tabanlı akıllı bot blok şeması

### 2.1. İlk Aşama: “Alma” Modülü

RAG tabanlı bu sistemde, Bağlam (B) olarak belirtilen girdiler, LLM’in eğitiminde kullanılmayan, İzmir Ekonomi Üniversitesi’ne ait eğitim ve öğretimle ilgili yönetmelik belgeleridir [8]. Sistemin amacı, öğrencilerin sorduğu soruları, bu yönetmelik belgesindeki bilgilere göre ve dil bilgisi kurallarına uygun olarak yanıtlamaktır.

HTML formatında ve Türkçe dilinde maddelerden oluşan bağlam, öncelikle veri işleme aşamasından geçirilmiştir. Bu aşamada, HTML’den TXT formatına çevirme, yazım hatalarının kontrolü, vb. işlemler yapılmış, her biri tek bir yönetmelik maddesinden oluşan M adet metin (B1, ..., BM) elde edilmiştir. Bu

metinlerin matematiksel olarak işlenebilmesi için sayısal formata dönüştürülmeleri gerekmektedir. Bu işlem, vektör gömme algoritması ile yapılmış ve

bağlam belgelerinin her bir maddesi,  $N$  boyutlu vektörlere dönüştürülmüştür  $(B_{g1}, \dots, B_{gM})$ .

**Tablo 1.** Literatürdeki benzer akıllı bot çalışmalarının önerilen yöntemle kıyaslaması

| Çalışma                                       | Yıl         | Dil            | Uygulama             | Yöntem (RAG/Diğer)                                                              |
|-----------------------------------------------|-------------|----------------|----------------------|---------------------------------------------------------------------------------|
| Ranoliya vd. [9]                              | 2017        | İngilizce      | Yönetmelikler        | Diğer                                                                           |
| Adams vd. [14], Nwankwo [11], Shivam vd. [17] | 2018        | İngilizce      | Yönetmelikler        | Diğer                                                                           |
| Krassman vd. [10]                             | 2019        | Portekizce     | E-öğrenme            | Diğer                                                                           |
| Chandra ve Suyanto [12]                       | 2019        | Endonezya dili | Yönetmelikler        | Diğer                                                                           |
| Gbenga vd. [15], Tommy vd. [13]               | 2020        | İngilizce      | Yönetmelikler        | Diğer                                                                           |
| Priadko [16]                                  | 2020        | Ukraynaca      | Yönetmelikler        | Diğer                                                                           |
| Mendoza vd. [18]                              | 2020        | İngilizce      | E-öğrenme            | Diğer                                                                           |
| Nguyen vd. [19]                               | 2021        | Vietnamca      | Yönetmelikler        | Diğer                                                                           |
| Nguyen vd. [25]                               | 2021        | Vietnamca      | E-öğrenme            | Diğer                                                                           |
| Lee vd. [29]                                  | 2022        | Tayvanca       | E-öğrenme            | Diğer                                                                           |
| Verma vd. [21]                                | 2022        | İngilizce      | Yönetmelikler        | RAG (kısmen)<br>Vektör gömme:<br>CountVectorizer<br>LLM: Yok                    |
| Ula vd. [20]                                  | 2023        | Fransızca      | E-öğrenme            | Diğer                                                                           |
| Ait Baha vd. [26]                             | 2023        | Fransızca      | E-öğrenme            | Diğer                                                                           |
| Aloqayli ve Abdelhafez [27]                   | 2023        | Arapça         | Yönetmelikler        | Diğer                                                                           |
| Alharethi [28]                                | 2023        | Arapça         | E-öğrenme            | Diğer                                                                           |
| Martinez-Araneda [30]                         | 2023        | İspanyolca     | E-öğrenme            | Diğer                                                                           |
| Oliveira ve Matos [22]                        | 2023        | İngilizce      | Yönetmelikler        | RAG<br>Vektör gömme: all-MiniLM-L6-v2<br>LLM: GPT-3                             |
| Maryamah vd. [23]                             | 2024        | İngilizce      | Yönetmelikler        | RAG<br>Vektör gömme: Ada<br>LLM: GPT-3.5 Turbo                                  |
| Neupane vd. [24]                              | 2024        | İngilizce      | Yönetmelikler        | RAG<br>Vektör gömme: text-embedding-3-large<br>LLM: GPT-3.5 Turbo               |
| <b>Önerilen yöntem</b>                        | <b>2025</b> | <b>Türkçe</b>  | <b>Yönetmelikler</b> | <b>RAG</b><br><b>Vektör gömme: text-embedding-3-large</b><br><b>LLM: GPT-4o</b> |

Bağlam vektörleri her soru-yanıt ikilisi için yeniden kullanılacakları için, gömme işlemini tekrarlamak zaman kaybına yol açacağından bu işlem yalnızca bir kere yapılmış ve bağlam vektörleri bir veri tabanında saklanmıştır. Kullanıcı bir soru sorduğunda ( $S$ ), bu soru da aynı vektör gömme algoritmasına girmiş ve soru vektörü ( $S_g$ ) elde edilmiştir. Vektör gömme işlemi için Türkçe destekli farklı kaynaklar denenmiş,

sonuçlar “Bulgular ve Yorumlar” bölümünde verilmiştir.

Bir sonraki aşamada, sorulan sorunun, bağlamın hangi maddeleri ile en çok ilişkili olduğu bulunmuştur. Bunun için, soru vektörü ( $S_g$ ) ile her bir bağlam maddesi vektörü ( $B_{gi}$ ) arasında benzerlik analizi yapılmıştır. Benzerlik kriteri olarak, en bilinen yöntem

olan kosinüs benzerliği seçilmiş ve vektörler arasındaki benzerlik ( $c_i$ ), Eşitlik (1)'deki gibi hesaplanmıştır:

$$c_i = \cos(S_g, B_{gi}) = \frac{\langle S_g, B_{gi} \rangle}{\|S_g\| \cdot \|B_{gi}\|} \quad \forall i = 1, \dots, M \quad (1)$$

Eşitlik (1)'deki  $\langle S_g, B_{gi} \rangle$  ifadesi  $S_g$  vektörü ile  $B_{gi}$  vektörünün iç çarpımını,  $\|S_g\|$  ve  $\|B_{gi}\|$  ifadeleri ise, sırasıyla  $S_g$  vektörü ile  $B_{gi}$  vektörünün Öklid uzaklığını göstermektedir. Hesaplanan  $c_i$  değeri ne kadar yüksekse  $B_{gi}$  vektörü de  $S_g$  vektörüne o kadar benzerdir. Bu da,  $B_i$  bağlam maddesinin  $S$  sorusuna anlamsal olarak yakınlığını temsil etmektedir. İlgili gömme vektörleri en yüksek “ $k$ ” kosinüs benzerliğini veren bağlam maddeleri ( $B'_{max}, B'_{max-1}, \dots, B'_{max-k+1}$ ) benzerlik analizinin çıktıları olarak seçilmiştir.

## 2.2 Son Aşama: “LLM Tabanlı Üretme” Modülü

Bu modülün girdileri, soru metni ( $S$ ) ile bu soruya anlamsal olarak en yakın, “alma” modülüne bulunmuş  $k$  adet bağlam metnidir ( $B'_{max}, B'_{max-1}, \dots, B'_{max-k+1}$ ). Bu girdiler, istem mühendisliği uygulanarak LLM'in daha iyi sonuç vermesi amacıyla zenginleştirilmiş, aşağıdaki gibi bir isteme dönüştürülmüştür:

*“Altındaki soruyu verilen bağlama göre yanıtla, sorunun yanıtı verilen bağlam içinde yoksa ‘bu soruyu yanıtlayacak yeterli bilgi yok’ yaz.*

*Bağlam:* [Bağlam]

*Soru:* [Soru]”

Hazırlanan bu istem, LLM'e girdi olarak verilmiştir. LLM seçiminde Türkçe desteği en önemli kriter olmuştur. Günümüzün en bilinen modellerinden, Türkçe'yi iyi anlayıp doğru yanıt verebilen OpenAI şirketinin GPT-4o modeli LLM olarak seçilmiştir.

LLM'in verdiği yanıt, en benzer bağlamı oluşturan yönetmelik maddeleriyle birlikte kullanıcıya sunulmuştur.

Önerilen sistemde, bağlama yeni belgeler ekleyerek kapsamı artırmak mümkündür. Bu işlem için, herhangi bir metin formatındaki yeni belgelerin maddelere ayrılması ve gömme algoritmasına sokularak vektör veri tabanında saklanması yeterlidir. Sistemin geri kalanında herhangi bir değişiklik yapılmasına gerek yoktur. Bu açıdan önerilen sistem, genişlemeye uygundur.

## III. BULGULAR VE TARTIŞMA

Bu çalışmada bağlam olarak İEÜ “Ön Lisans ve Lisans Eğitim-Öğretim ve Sınav Yönetmeliği” [8] kullanılmıştır. Bu doküman, 49 maddeden oluşmaktadır. Çoğu maddenin alt bentleri de

bulunmaktadır. Vektör gömme işleminden önce, her bir bent bir paragraf olacak şekilde 183 parçaya ayrılmıştır ( $M = 183$ ). Örnek bir paragraf şu şekildedir:

*“MADDE 18. Ders yükü. (1) Bir öğrencinin her dönemde alacağı normal ders yükü 30 AKTS kredisidir. Tekrarlanan dersler normal ders yükünden sayılır.”*

### 3.1 Vektör Gömme Bulguları

Metinlerin sayısal biçime dönüştürülmesi işlemi olan vektör gömmenin kalitesinin, RAG sisteminin başarımındaki etkisi büyüktür. Bu nedenle bu işlem dil kurallarına uygun olarak yapılmalıdır ve kullanılacak olan yazılım programı, Türkçe metinleri Türkçe dil kurallarına göre sayısal vektörlere çevirmelidir. Fakat Türkçe destekli vektör gömme yazılımları, İngilizce için olanlar kadar fazla ve yaygın değildir. İlk olarak, profesyonel modeller arasında en bilinen ve Türkçe'de en başarılı sonuçlar veren OpenAI şirketinin “text-embedding-3-large” modelinin gömme algoritması seçilmiştir (Algoritma-1). Bu model, metinleri en fazla 8191 token sayısına göre, boyutu  $N = 3072$  olan vektörlere çevirmektedir.

Kıyaslama yapabilmek için bir de Hugging Face portalı taranmış ve en çok kullanılan Türkçe destekli, BERT tabanlı “bert-base-turkish-cased-mean-nli-stsb-tr” modeli kullanılmıştır (Algoritma-2). Bu model ile, en fazla 512 token sayısına göre,  $N = 768$  boyutlu sayısal vektörler elde edilmiştir.

Her iki gömme modelini kullanırken, vektörlere dönüştürülen metinlerin token sayıları kontrol edilmiştir. Buradaki amaç, token sınırını aşan metin varsa onu bulup daha küçük parçalara bölmektir. Ne var ki kullanılan metinler yönetmelik maddeleri oldukları için kısa paragraflardan oluşmaktadırlar ve token sınırının içinde kalmaktadırlar. Algoritma-1'in ürettiği en küçük tokenli madde için token sayısı 31, en büyüğü için 618'dir. Algoritma-2 ise en küçük tokenli madde için 19 token, en büyüğü için 377 token kullanmıştır. Buradan, Algoritma-1'in Türkçe için daha fazla çeşitli token kullandığı anlaşılmaktadır.

Bu iki vektör gömme modeli ile ilgili bulgular bir sonraki bölümde, benzerlik analizi bulguları ile birlikte verilmiştir. Her iki modeli kullanmak için Python dilinde kod yazılmış ve modellere API yoluyla ulaşılmıştır.

### 3.2 Benzerlik Analizi Bulguları

RAG sistemindeki “alma” modülünün bağlamdaki yönetmelik maddelerinden sorulan soruya kosinüs benzerliği kriteriyle en benzer kaç tanesini almak gerektiği ile ilgili bir formül yoktur, buradaki  $k$  bir hiper-parametredir. Küçük bir  $k$  değeri kullanmak, az

sayıda madde ile soruyu yanıtlamaya çalışmak demek olacaktır ki, bazı durumlarda yeterli bağlam verilmediği için sorunun yanıtsız kalmasına neden olabilir. Öte yandan büyük bir  $k$  değeri kullanmak, hem benzerlik analizinde gereksiz yere zaman kaybetmeye hem de LLM'e gereğinden uzun bir bağlam verip odağı dağıtmaya ya da fazla ücret ödemeye yol açabilir. Dolayısıyla en iyi  $k$  değerini bulmak önemlidir.

Örneğin, “not ortalamam 2,35 bu dönem ilave kaç AKTS alabilirim?” sorusunun yanıtı, yönetmeliğin 18. maddesinin 2. bendinde yazmaktadır:

*MADDE 18. Ders yükü. (2) Öğrenci normal ders yüküne ilave olarak; a) Genel not ortalaması 2,00 ve 2,49 arasında olan öğrenciler için; öğrencinin talebi ve danışmanın onayı ile toplamda 6 AKTS kredi, b) Genel not ortalaması en az 2,50 olan öğrenciler için öğrencinin talebi ve danışmanın onayı ile toplamda 12 AKTS kredi alabilirler.*

Algoritma-1 ile oluşturulan gömme vektörleri üzerinden yapılan benzerlik analizinde en benzer 10 madde şunlar çıkmıştır:

Madde 26 (2), Geçici Madde 1 (2), Madde 18 (2), Madde 27 (1), Madde 18 (1), Madde 27 (2), Madde 24 (1), Madde 17 (3), Madde 3 (1c), Madde 26 (3).

Görüldüğü gibi, yanıtın olduğu Madde 18 (2), üçüncü sıradadır. Fakat Algoritma-2 ile oluşturulan gömme vektörleri üzerinden yapılan benzerlik analizinde, aynı soru için sıralama şu şekilde çıkmıştır:

Madde 24 (2), Madde 26 (2), Madde 23 (8), Madde 24 (3c), Madde 24 (3f), Madde 28 (1b), Madde 28 (1a), Madde 27 (1), Madde 18 (2).

Sıralamadan görüldüğü gibi, yanıtı içeren Madde 18 (2), 10. sırada yer almıştır. Bu örnek, Algoritma-1'in, Algoritma-2'ye kıyasla Türkçe'ye daha uygun gömme vektörler ürettiğini göstermektedir.

Sonucu genelleştirebilmek için, yönetmeliğin her maddesi ile ilgili en az bir tane olmak üzere, yukarıdaki örneğe benzer 100 adet soru hazırlanmıştır. Daha sonra farklı  $k$  değerleri için her soru ile, her iki gömme modelinin vektörlerini kullanarak benzerlik analizi yapılmış ve bulunan yönetmelik maddeleri kontrol edilmiştir. Her  $k$  değerinde, 100 sorudan kaç tanesi için doğru maddenin en benzer  $k$  madde içinde olduğu Tablo 2'de verilmiştir.

**Tablo 2.** Farklı gömme vektörleri ile benzerlik analizi sonuçları

| En benzer $k$ madde | Algoritma-1 | Algoritma-2 |
|---------------------|-------------|-------------|
| 1                   | 69          | 61          |
| 2                   | 82          | 74          |
| 3                   | 88          | 83          |
| 4                   | 92          | 85          |
| 5                   | <b>100</b>  | 90          |
| 6                   | 100         | 92          |
| 7                   | 100         | 96          |
| 8                   | 100         | 97          |
| 9                   | 100         | 98          |
| 10                  | 100         | <b>100</b>  |

Tablo 2'de görüldüğü gibi, Algoritma-1'in gömme vektörleri ile alınan sonuçlarda,  $k \leq 4$  için en benzer bulunan maddeler içinde doğru yanıtı içeren maddenin olduğu durumlar %100'den azdır. Ancak  $k = 5$  için tüm sorulara doğru yanıtı içeren maddeler bulunmuştur. Bu, tüm sorular için en çok  $k = 5$  maddeye bakmak yeterli demektir.

Yine Tablo 2'ye bakıldığında, Algoritma-1'in gömme vektörleri ile alınan sonuçların Algoritma-2'nin ürettiği vektörlerle alınan sonuçlardan daha iyi olduğu görülmektedir. Algoritma-1, beş yönetmelik maddesi içinde doğru yanıtı içeren maddeyi bulurken Algoritma-2 ile tüm sorulara doğru yanıt verebilmek için en benzer 10 maddeyi kullanmak gerekmektedir.

Burada vektör gömme modeli ile ilgili seçim yapılırken, sonuçların doğruluğunun yanı sıra dikkate alınması gereken diğer konu, Algoritma-1 ücretli iken (Eylül 2024 itibarı ile \$0.00013 / 1k token) Algoritma-2'nin ücretsiz olmasıdır.

Bu bölümdeki bulgulara ulaşmak için de yine Python dili kullanılmış ve gömme vektörlerinin kosinüs benzerliklerini bulmak için kod yazılmıştır.

### 3.3 LLM Bulguları

Bir önceki bölümde verilen vektör gömme ve benzerlik analizi sonuçları değerlendirilerek, bundan sonraki LLM denemelerine, Algoritma-1 ve  $k = 5$  kosinüs benzerliği kullanılarak devam edilmiştir. Böylece, “alma” modülünün %100 doğru çalışacağı garanti edilmiştir.

LLM değerlendirme ölçütü olarak literatürde ROUGE [32] ve BLEU [33] gibi popüler yöntemler bulunmaktadır. Fakat, ROUGE genel olarak özet çıkarma, BLEU ise çeviri işlerini değerlendirmek için kullanıldığı için bu çalışmanın bağlamına uygun değildir. Ayrıca bu stratejilerin dayandıkları LLM içi sözcük olasılıklarına GPT-4o gibi kapalı sistemlerde erişim mümkün değildir.

Bu nedenlerle bu çalışmada kullanılan LLM olan GPT-4o modelini değerlendirmek üzere Zhang vd.'nin kullandığına yakın, kosinüs benzerliğine dayalı bir ölçüt kullanılmıştır [34]. Bunun için daha önce benzerlik analizinde kullanılan 100 soru burada da ele alınmıştır. Her soru ve en benzer beş bağlam maddesi isteme dönüştürülerek LLM'e verilmiş ve yanıt alınmıştır. LLM'in yanıtının ve sorulan sorunun yanıtını içeren bağlam maddesinin Algoritma-1 ile

gömme vektörleri oluşturulmuş; bu vektörlerin kosinüs benzerliği hesaplanmıştır. Sonuçlardan bir kesit Tablo 3'te verilmiştir.

Tablo 3'te görüldüğü gibi, LLM'e farklı tipte sorular yöneltilmiştir. Sorulan 100 sorudan 57'si birinci sorudaki gibi ucu açık, 31'i ikinci sorudaki gibi olumlu, 12'si 100. sorudaki gibi olumsuz yanıtıdır.

**Tablo 3.** LLM yanıt örnekleri

| No       | Soru                                                                        | Yanıtı içeren bağlam                                                                                                                                                                                                                                                                                                                                                                                                                                           | LLM yanıtı                                                                                                                                                                                                         | Benzerlik |
|----------|-----------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| 1        | Azami öğrenim süresi ne kadar?                                              | MADDE 30. Öğrenim süresi. (2) Azami eğitim öğretim süresi, ön lisans programları için dört yıl (sekiz yarıyıl); lisans programları için yedi yıl (on dört yarıyıl), Tıp Fakültesi için dokuz yıl (on sekiz yarıyıl)'dır.                                                                                                                                                                                                                                       | Azami eğitim öğretim süresi, ön lisans programları için dört yıl (sekiz yarıyıl), lisans programları için yedi yıl (on dört yarıyıl), Tıp Fakültesi için dokuz yıl (on sekiz yarıyıl) olarak belirlenmiştir.       | 0,98      |
| 2        | Notuma itiraz edebilir miyim?                                               | MADDE 25. Notlarda maddi hata. (1) Ön lisans ve lisans programlarında yer alan herhangi bir dersin değerlendirme sonucuna; ilanı tarihinden itibaren en geç üç iş günü içinde itiraz edilebilir. Öğrenci Bilgi Sistemi üzerinden açıklanan dönem sonu ders notları ile ilişkili herhangi bir maddi hatanın yapılmış olması halinde, düzeltme istemi, öğretim elemanının başvurusu üzerine ilgili Fakülte-Yüksekokul Yönetim Kurulu tarafından karara bağlanır. | Evet, itiraz edebilirsiniz. Ön lisans ve lisans programlarında yer alan herhangi bir dersin değerlendirme sonucuna, ilanı tarihinden itibaren en geç üç iş günü içinde itiraz edebilirsiniz.                       | 0,81      |
| ...      |                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                    |           |
| 100      | Dersler başladıktan sonra kaydımı sildirirsem ücretimi geri alabilir miyim? | MADDE 34. Ücretler. (4) Üniversiteye yeni kayıt olan öğrencilerden veya Üniversiteye devam eden öğrencilerden kayıt sildirme başvurusunu akademik takvimde ilan edilen derslerin başlangıç tarihine kadar yapanlara, ödedikleri öğrenim ücreti üzerinden yıllık öğrenim bedelinin %15'i kesilerek kalan miktar iade edilir. Derslerin başlangıç tarihinden sonra kayıt sildirme başvurusunu yapan öğrencilerin ise ödemiş oldukları ücretler iade edilmez.     | Derslerin başlangıç tarihinden sonra kayıt sildirme başvurusunu yapan öğrencilerin ödemiş oldukları ücretler iade edilmez. Bu nedenle, dersler başladıktan sonra kaydınızı sildirirseniz, ücreti iade alamazsınız. | 0,71      |
| Ortalama |                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                    | 0,82      |

Böylece, LLM'in farklı tipte sorulara nasıl karşılık vereceği sınanmıştır. Sözü geçen 100 soru ile, bağlamı oluşturan yönetmeliğin her maddesi kapsanmıştır. Dolayısıyla soru sayısı yeterli görülmüştür.

Tüm sorular için, yanıtı içeren bağlam (sütun 3) ile LLM yanıtı (sütun 4) arasındaki benzerlik değerlerinin (sütun 5) ortalaması 0,82 çıkmıştır. Bu başarı çok yüksek görünmese de yanıtlar bir insan tarafından değerlendirdiğinde LLM'in tüm sorulara doğru yanıt verdiği anlaşılmaktadır. Son sorudaki gibi çok yüksek olmayan benzerlik oranının (0,71) nedeni, bağlamın sorulan sorudan daha çok bilgi içermesi, LLM'in bu

bağlamdan sadece sorunun yanıtını çıkarmasıdır. Dolayısıyla, bağlam ile yanıt benzerliğine bakıldığında oran çok yüksek çıkmamıştır; fakat sorunun yanıtı doğru olarak verilmiştir. Bu açıdan ortalama 0,82 benzerlik LLM performansı açısından başarılı kabul edilmelidir.

Ayrıca bağlamda olmayan bir soru sorulduğunda LLM'in yanlış bir yanıt vermediği de sınanmıştır. Örneğin, "üniversitenin en başarılı bölümü hangisi?" sorusuna "bu soruyu yanıtlamak için yeterli bilgim yok" şeklinde dönüş yapmıştır. Bu soruyla ilgili bir bilgi

yönetmelik maddelerinde bulunmadığı için bu doğru bir tutumdur.

Bu bölümdeki denemeler için de Python dilinde kod yazılmış ve LLM'e API yoluyla ulaşılmıştır.

### 3.4 EkoBot Arayüzü

Önerilen sistemin örnek kullanımı için, Şekil 2'deki gibi bir Web arayüzü tasarlanmıştır. Kullanıcı sorusunu metin olarak girdiğinde hem LLM'in yanıtını hem de bu yanıtı oluşturmak için yararlanılan yönetmelik maddelerini görebilmektedir.

Önceki bölümlerde anlatıldığı üzere, bu prototipte bağlam olarak en benzer beş ( $k = 5$ ) madde kullanılmıştır.

Şekil 2'de görüldüğü gibi, kullanıcı her yanıt için, emojiyle ifade edilen, iyi, orta, kötü olmak üzere üç tip geribildirim verebilmektedir. Ayrıca, metin olarak yorumunu da yazabilmektedir. Bu arayüzde yanıtın yanı sıra yararlanılan kaynakların da gösterilmesi, kullanıcının bilgilendirilmesi açısından yararlı bulunmuştur.

Şekil 2. Örnek EkoBot Web arayüzü

## IV. SONUÇ

Bu makale, Türkçe desteği olan sanal akademik danışman EkoBot'un geliştirilmesi ve sınanması üzerine kapsamlı bir çalışmayı sunmaktadır. Bu bot, RAG yöntemiyle çalışmakta, LLM olarak GPT-4o'yu kullanmaktadır.

RAG yönteminde “alma” ve “LLM tabanlı üretme” modülleri bulunmaktadır. İzmir Ekonomi Üniversitesi'nin “Ön Lisans ve Lisans Eğitim-Öğretim ve Sınav Yönetmeliği” belgesi içerisinden, sorulan soruya en yakın beş maddeyi seçme görevini “alma” modülü yerine getirmektedir ve bu seçimi yaparken gömülü vektörlerin kosinüs benzerliğinden yararlanmaktadır. Yönetmelik maddelerinin gömülü vektörlere dönüştürülmesi denemelerinde en iyi sonucu OpenAI “text-embedding-3-large” algoritması vermiştir.

İstem mühendisliği yöntemi kullanılarak, sorulan soru ve soruya en benzer beş yönetmelik maddesi LLM'e verilmiş ve yanıt alınmıştır. Bu işlem 100 soruluk bir veri kümesi üzerinden yapılmış ve LLM'in her soruya doğru yanıt verdiği gözlemlenmiştir. Yanıtlar ve yanıt

içeren yönetmelik maddeleri arasındaki ortalama kosinüs benzerliği 0,82 olarak hesaplanmıştır.

Önerilen sistem, bağlam belgelerinin artırılmasına ve kullanıcıdan gelecek geri bildirimlere göre yeni bağlam belgeleri eklenmesine uygundur. Bu işlem, LLM'in yeniden eğitilmesini gerektirmediğinden, yalnızca veri işleme aşamasından oluşacağından düşük maliyet ile yapılabilecektir. Bu uygulamanın ilk olarak test sürümü biçiminde öğrencilere açılması planlanmaktadır. Böylece, öğrencilerden gelecek sorulara göre veri kümesindeki eksiklikler belirlenecek ve gerekli yeni bağlam belgeleri sisteme eklenecektir.

OpenAI şirketinin sunduğu gibi, vektör gömme algoritmalarına ya da LLM'lere API üzerinden uzaktan erişim sayesinde, yerel bilgisayarda LLM çalıştırma, dolayısıyla da güçlü işlemci gücü gereksinimi ortadan kalkmakta, yalnızca İnternet bağlantısı gerekmektedir. Bu modellerin kullanım ücretleri ise makul sınırlar içerisindedir ve günden güne azalmaktadır.

Burada önerilen çözüm sayesinde öğrencilere zaman sınırı olmaksızın doğru bilgiler verilebilecek, süreç otomatikleştirilerek iyileştirilecek ve öğrenci

memnuniyeti arttırılacaktır. Bu açılardan, üniversite ortamı için Türkçe destekli akıllı sanal danışman botu yararlı bir uygulamadır.

Bundan sonrasında, vektör gömme işlemi için Google Embedding-001 vb. gibi başka algoritmalar, benzerlik hesapları için maksimum marjinal alaka (Maximal Marginal Relevance, MMR), Öklid uzaklığı vb. gibi başka yöntemler, LLM için Google Gemini vb. gibi başka büyük dil modelleri denenebilir. Bot arayüzü geliştirilirken sohbet akışı şeklinde hafıza eklenebilir. Yazılım kısmını kolaylaştırmak için LangChain, AgentGPT vb. gibi altyapı platformları kullanılabilir.

Öğrenci İşleri Müdürlüğü gibi üniversite idari birimlerinin verdiği geri beslemeye göre, sınav yönetmelikleri ve “sıkça sorulan sorular” gibi bilgiler, üniversite Web sayfalarında bulunmasına rağmen, öğrenciler ya bu sayfaların yerini bulamamakta, ya aradıkları bilginin hangi yönetmelikte olduğunu bilememekte, ya da sayfa içinde arama yaparken kullandıkları sözcükler metinlerde aynı biçimde yer almadığı için bilgiye erişememektedirler. Dolayısıyla öğrencilerin doğrudan ulaşabilecekleri ve sordukları soruları farklı ifadelerle de olsa anlayıp doğru bilgilere göre yanıtlayabilen bir RAG sistemi, var olan yaklaşımlara kıyasla daha kullanışlı ve verimlidir. Bu nedenle önerilen bu sistem, hem öğrencilere, hem akademisyenlere, hem de idari çalışanlara kolaylık sağlamaktadır.

## TEŞEKKÜR

Bu çalışma, OpenAI API Researcher Access Program tarafından API kredisi ile desteklenmiştir.

## KAYNAKLAR

- [1] Ayanouz, S., Abdelhakim, B. A., & Benhmed, M. (2020). A Smart Chatbot Architecture based NLP and Machine Learning for Health Care Assistance. *Proceedings of the 3rd International Conference on Networking, Information Systems & Security*, 1–6. <https://doi.org/10.1145/3386723.3387897>
- [2] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *Proceedings of the 34th International Conference on Neural Information Processing Systems (NIPS '20)*, 9459–9474.
- [3] Lee, K., Chang, M.-W., & Toutanova, K. (2019). Latent Retrieval for Weakly Supervised Open Domain Question Answering. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 6086–6096. <https://doi.org/10.18653/v1/P19-1612>
- [4] Guu, K., Lee, K., Tung, Z., Pasupat, P., & Chang, M.-W. (2020). Retrieval-Augmented Language Model Pre-Training. *Proceedings of Machine Learning Research*, 3929–3938.
- [5] Khattab, O., Santhanam, K., Li, X. L., Hall, D., Liang, P., Potts, C., & Zaharia, M. (2022). *Demonstrate-Search-Predict: Composing retrieval and language models for knowledge-intensive NLP*.
- [6] Ram, O., Levine, Y., Dalmedigos, I., Muhlga, D., Shashua, A., Leyton-Brown, K., & Shoham, Y. (2023). *In-Context Retrieval-Augmented Language Models*.
- [7] Shi, W., Min, S., Yasunaga, M., Seo, M., James, R., Lewis, M., Zettlemoyer, L., & Yih, W. (2023). *REPLUG: Retrieval-Augmented Black-Box Language Models*.
- [8] İEÜ Ön Lisans ve Lisans Eğitim-Öğretim ve Sınav Yönetmeliği. (2024). <https://www.ieu.edu.tr/tr/bylaws/type/read/id/72>.
- [9] Ranoliya, B. R., Raghuwanshi, N., & Singh, S. (2017). Chatbot for university related FAQs. *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 1525–1530. <https://doi.org/10.1109/ICACCI.2017.8126057>
- [10] Krassmann, A. L., Flach, J. M., Grand, A. R. C. da S., Tarouco, L. M. R., & Bercht, M. (2019). A Process for Extracting Knowledge Base for Chatbots from Text Corpora. *2019 IEEE Global Engineering Education Conference (EDUCON)*, 322–329. <https://doi.org/10.1109/EDUCON.2019.8725064>
- [11] Nwankwo, W. (2018). Interactive Advising with Bots: Improving Academic Excellence in Educational Establishments. *American Journal of Operations Management and Information Systems*, 3(1), 6. <https://doi.org/10.11648/j.ajomis.20180301.12>
- [12] Chandra, Y. W., & Suyanto, S. (2019). Indonesian Chatbot of University Admission Using a Question Answering System Based on Sequence-to-Sequence Model. *Procedia Computer Science*, 157, 367–374. <https://doi.org/10.1016/j.procs.2019.08.179>
- [13] Tommy, L., Kirana, C., & Riska, L. (2020). The Combination of Natural Language Processing and Entity Extraction for Academic Chatbot. *2020 8th International Conference on Cyber and IT Service Management (CITSM)*, 1–6. <https://doi.org/10.1109/CITSM50537.2020.9268851>
- [14] Guvindan Raju, K. R., Adams, C., & Srinivas, R. (2018). Cognitive Virtual Admissions Counselor. *SMU Data Science Review*, 1(1), 1–13.
- [15] Gbenga, O., Okedigba, T., & Oluwatobi, H. (2020). An Improved Rapid Response Model for University Admission Enquiry System Using Chatbot. *International Journal of Computer (IJC)*, 38(1), 123–131.

- [16] Priadko, A. O., Osadcha, K. P., Kruhlyk, Vladyslav S., & Rakovych, V. A. (2019). Development of a chatbot for informing students of the schedule. *Proceedings of the 2nd Student Workshop on Computer Science & Software Engineering (CS&SE@SW 2019)*, 128–137.
- [17] Shivam, K., Saud, K., Sharma, M., Vashishth, S., & Patil, S. (2018). Chatbot for College Website. *International Journal of Computing and Technology*, 5(6), 74–77.
- [18] Mendoza, S., Hernández-León, M., Sánchez-Adame, L. M., Rodríguez, J., Decouchant, D., & Meneses-Viveros, A. (2020). Supporting Student-Teacher Interaction Through a Chatbot. In *Learning and Collaboration Technologies. Human and Technology Ecosystems* (pp. 93–107). Springer International Publishing. [https://doi.org/10.1007/978-3-030-50506-6\\_8](https://doi.org/10.1007/978-3-030-50506-6_8)
- [19] Nguyen, T. T., Le, A. D., Hoang, H. T., & Nguyen, T. (2021). NEU-chatbot: Chatbot for admission of National Economics University. *Computers and Education: Artificial Intelligence*, 2, 100036. <https://doi.org/10.1016/j.caeai.2021.100036>
- [20] Ula, M., Hardi, R., & Hipiny, I. (2023). An Improved Structure for Academic Information Services through AI Chatbots. *Journal of Engineering Science and Technology Review*, 16(5), 164–173. <https://doi.org/10.25103/jestr.165.20>
- [21] Verma, A., Kuntala, C., Khatri, P., . S., Kaur, S., Mohapatra, A. K., & Singhal, S. (2022). University Chatbot System Using NLP. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.4255753>
- [22] Oliveira, P. F., & Matos, P. (2023). Introducing a Chatbot to the Web Portal of a Higher Education Institution to Enhance Student Interaction. *ASEC 2023*, 128. <https://doi.org/10.3390/ASEC2023-16621>
- [23] Maryamah, M., Irfani, M. M., Tri Raharjo, E. B., Rahmi, N. A., Ghani, M., & Raharjana, I. K. (2024). Chatbots in Academia: A Retrieval-Augmented Generation Approach for Improved Efficient Information Access. *2024 16th International Conference on Knowledge and Smart Technology (KST)*, 259–264. <https://doi.org/10.1109/KST61284.2024.10499652>
- [24] Neupane, S., Hossain, E., Keith, J., Tripathi, H., Ghiasi, F., Golilarz, N. A., Amirlatifi, A., Mittal, S., & Rahimi, S. (2024). From Questions to Insightful Answers: Building an Informed Chatbot for University Resources. *ArXiv*.
- [25] Nguyen, H. D., Tran, T.-V., Pham, X.-T., Huynh, A. T., & Do, N. V. (2021). Ontology-based Integration of Knowledge Base for Building an Intelligent Searching Chatbot. *Sensors and Materials*, 33(9), 3101. <https://doi.org/10.18494/SAM.2021.3264>
- [26] Ait Baha, T., El Hajji, M., Es-Saady, Y., & Fadili, H. (2024). The impact of educational chatbot on student learning experience. *Education and Information Technologies*, 29(8), 10153–10176. <https://doi.org/10.1007/s10639-023-12166-w>
- [27] Aloqayli, A., & Abdelhafez, H. (2023). Intelligent Chatbot for Admission in Higher Education. *International Journal of Information and Education Technology*, 13(9), 1348–1357. <https://doi.org/10.18178/ijiet.2023.13.9.1937>
- [28] Alharethi, T. M. (2023). Autoresponder using Chatbot for Educational Services. *2023 1st International Conference on Advanced Innovations in Smart Cities (ICAISC)*, 1–5. <https://doi.org/10.1109/ICAISC56366.2023.10084956>
- [29] Lee, Y.-F., Hwang, G.-J., & Chen, P.-Y. (2022). Impacts of an AI-based chabot on college students' after-class review, academic performance, self-efficacy, learning attitude, and motivation. *Educational Technology Research and Development*, 70(5), 1843–1865. <https://doi.org/10.1007/s11423-022-10142-8>
- [30] Martinez-Araneda, C., Gutiérrez, M., Maldonado, D., Gómez, P., Segura, A., & Vidal-Castro, C. (2024). *Designing a chatbot to support problem-solving in a programming course*. 966–975. <https://doi.org/10.21125/inted.2024.0317>
- [31] Es, S., James, J., Espinosa-Anke, L., & Schockaert, S. (2023). RAGAS: Automated Evaluation of Retrieval Augmented Generation. *ArXiv*.
- [32] Lin, C.-Y. (2004). ROUGE: A Package for Automatic Evaluation of Summaries. *Text Summarization Branches Out*, 74–81.
- [33] Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2001). BLEU. *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02*, 311. <https://doi.org/10.3115/1073083.1073135>
- [34] Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., & Artzi, Y. (2020, April 26). BERTScore: Evaluating Text Generation with BERT. *8th International Conference on Learning Representations*.