Faculteit Industriële Ingenieurswetenschappen

master in de industriële wetenschappen: informatica

Masterthesis

Ontwikkeling van een AI-gebaseerd prototype dat studenten en docenten ondersteunt met oefeningen

Jarne Thys

Scriptie ingediend tot het behalen van de graad van master in de industriële wetenschappen: informatica

PROMOTOR :
Prof. dr. Davy VANACKEN

BEGELEIDER :
De heer Sebe VANBRABANT

COPROMOTOR :
Prof. dr. Gustavo Alberto ROVELO RUIZ

Gezamenlijke opleiding UHasselt en KU Leuven

Universiteit Hasselt | Campus Diepenbeek | Faculteit Industriële Ingenieurswetenschappen | Agoralaan Gebouw H - Gebouw B | BE 3590 Diepenbeek

Universiteit Hasselt | Campus Diepenbeek | Agoralaan Gebouw D | BE 3590 Diepenbeek
Universiteit Hasselt | Campus Hasselt | Martelarenlaan 42 | BE 3500 Hasselt

2023
2024

Faculteit Industriële Ingenieurswetenschappen

master in de industriële wetenschappen: informatica

Masterthesis

Ontwikkeling van een AI-gebaseerd prototype dat studenten en docenten ondersteunt met oefeningen

Jarne Thys

Scriptie ingediend tot het behalen van de graad van master in de industriële wetenschappen: informatica

PROMOTOR :
Prof. dr. Davy VANACKEN

COPROMOTOR :
Prof. dr. Gustavo Alberto ROVELO RUIZ

BEGELEIDER :
De heer Sebe VANBRABANT

UHASSELT KU LEUVEN

Woord vooraf

Mijn oprechte dank gaat uit naar mijn promotor, prof. dr. Davy Vanacken, en copromotor, prof. dr. Gustavo Rovelo Ruiz, voor hun waardevolle begeleiding, deskundige feedback en directe beschikbaarheid. De ondersteuning van Sebe Vanbrabant was eveneens van onschatbare waarde bij het schrijven van deze masterproef en het ontwikkelen van het prototype. Hun betrokkenheid heeft mijn onderzoek versterkt en dit traject mogelijk gemaakt.

Inhoudsopgave

Lijst van tabellen

Lijst van figuren

Verklarende woordenlijst

API	Application Programmable Interface; Een API biedt een gestructureerde manier voor programma's om gegevens uit te wisselen en acties uit te voeren zonder dat de interne werking van elkaar volledig bekend hoeft te zijn.
CPU	Central Processing Unit; Het brein van een computer dat verantwoordelijk is voor het uitvoeren van instructies en taken binnen een computerprogramma door data te verwerken en berekeningen uit te voeren.
GPU	Graphics Processing Unit; Een gespecialiseerde chip die verantwoordelijk is voor het versnellen van grafische taken en het renderen van afbeeldingen, video's en 3D-graphics op een computer. Ook vaak gebruikt om machine learning taken te versnellen.
RAM	Random Access Memory; Een vorm van computergeheugen dat tijdelijk data en instructies opslaat die op dat moment door de processor worden gebruikt. Het stelt de computer in staat om snel toegang te krijgen tot informatie die nodig is voor lopende taken, waardoor het de prestaties verbetert door snelle gegevenstoegang en -verwerking mogelijk te maken.
Prompt	Een instructie of input die wordt gebruikt om een reactie te genereren Large Language Models. Het is de input die aangeeft wat je van het systeem wilt ontvangen.

Abstract

Met de opkomst van Large Language Models (LLM's) is er nu een mogelijkheid voor studenten om vragen te stellen aan artificiële intelligentie (AI). Er zijn echter zorgen onder docenten over de invloed van AI op de onderwijskwaliteit, bijvoorbeeld door de nauwkeurigheid van door AI gegenereerde antwoorden en mogelijks verminderde interactie met studenten. Deze masterproef focust op het verkennen van de toepassingen en uitdagingen van AI in het onderwijs, en vooral dan LLM's, en het creëren van een prototype dat demonstreert hoe deze toepassingen kunnen worden geïmplementeerd.

Na het literatuuronderzoek worden enkele experimenten uitgevoerd waarbij de antwoorden van LLM's worden voorgelegd aan docenten. De bevindingen hieruit dienen als basis voor het ontwikkelen van verschillende tools die docenten en studenten kunnen ondersteunen bij oefeningen. Uiteindelijk wordt er een applicatie ontwikkelt die de verschillende tools beschikbaar maakt op een manier die transparant is over de gebruikte AI-tools.

Door de docent aanpasbare LLM's, oefeningen voorgesteld door een recommender en een dynamische FAQ dragen bij aan de mogelijkheden om het onderwijs te verbeteren met AI. Verzamelde data zorgen ervoor dat de interactie tussen docenten en studenten niet verminderd en docenten hun materiaal beter kunnen afstemmen op de studenten. Dit is een kleine greep uit de mogelijkheden van AI in het onderwijs. Verder onderzoek kan zich richten op het uitvoeren van een user study om de effectiviteit en gebruikservaring te evalueren.

Abstract in English

With the emergence of Large Language Models (LLMs), there is now an opportunity for students to ask artificial intelligence (AI) questions. However, there are concerns among teachers about the impact of AI on teaching quality, for example due to the accuracy of AI-generated answers and potentially reduced interaction with students. This master's thesis focuses on exploring the applications and challenges of AI in education, especially LLMs, and creating a prototype that demonstrates how these applications can be implemented.

Following the literature review, some experiments are conducted in which LLMs' responses are presented to teachers. The findings from these serve as the basis for developing several tools that can support teachers and students in exercises. Ultimately, an application is developed that makes the different tools available in a way that is transparent about the AI tools used.

Teacher-customisable LLMs, exercises suggested by a recommender and a dynamic FAQ add to the possibilities of improving teaching with AI. Collected data ensures that the interaction between teachers and students does not diminish and teachers can better tailor their material to students. This is a small sample of the possibilities of AI in education. Further research can focus on conducting a user study to evaluate the effectiveness and user experience.

Hoofdstuk 1

Inleiding

1.1 Situering

Artificiële intelligentie (AI) is een onderwerp dat bijzonder populair is. Met een wereldwijde marktomzet van 12,6 miljard USD in 2022 zijn ook veel bedrijven geïnteresseerd in AI (Valuates, 2023a). Large Language Models (LLM's) zijn een recente ontwikkeling binnen AI die de AI-markt nog een extra boost gegeven hebben. LLM's waren goed voor een marktomzet van 10,5 miljard USD in 2022 en wordt verwacht te stijgen tot 40,8 miljard USD in 2029 (Valuates, 2023b). Deze nieuwe technologie werd wereldwijd bekent met de introductie van ChatGPT in november 2022 (OpenAI, 2022). Gebruikers kunnen met deze chatbot gesprekken voeren alsof er een echte persoon voor hen zit. Voordat LLM's gebruikt werden waren al chatbots die natuurlijke zinnen konden begrijpen en ook in een natuurlijke taal konden antwoorden. LLM's doen hier echter nog een schepje bovenop doordat ze niet specifiek voor een onderwerp getraind zijn. Met andere chatbots in het verleden was dit wel het geval, waardoor hun inzetbaarheid eerder specifiek en beperkt was. LLM's zijn getraind op grote hoeveelheden data waardoor ze over vrijwel elk onderwerp kunnen meespreken (Radford et al., 2019). Figuur 1.1 toont een voorbeeld van een gesprek met ChatGPT-3.5.

De opkomst van zulke nieuwe technologie opent veel deuren voor nieuw onderzoek op basis van deze technologie. Ook in het Expertisecentrum voor Digitale Media (EDM), een onderzoekscentrum van de Universiteit Hasselt, wordt al verschillende jaren onderzoek naar artificiële intelligentie gedaan (EDM, 2024a). Als onderdeel van het Vlaamse AI-onderzoeksprogramma staan zij mee aan de basis van baanbrekend AI-onderzoek (Flanders AI, 2024). Zij specialiseren zich onder meer in *Intelligible Interactive Systems*. Dit betekent dat de intelligente systemen die ze ontwikkelen ook verklaarbaar en transparant moeten zijn voor de gebruiker (EDM, 2024b). Als onderdeel van Flanders Make doet het EDM ook veel onderzoek voor de maakindustrie. Hierin zijn ze onder andere bezig met de toepassing van AI ter verbetering van digitale werkinstructies en het welzijn van operatoren.

Naast dergelijke toepassingen, heeft de opkomst van diverse vormen van artificiële intelligentie ook een impact op het onderwijslandschap. Studenten kunnen gebruik maken van LLM's om complexe teksten te genereren. Hierdoor wordt, bijvoorbeeld, het schrijven van een paper veel makkelijker (Ceuppens, 2023). Er zijn veel vragen over de legaliteit van het gebruiken van zulke tools. De LLM's zijn getraind op werken die auteursrechtelijk beschermd kunnen zijn,

Anonymous

Gegeven is de volgende reeks: {2, 4, 6, 8, 10, 12, 13}. Hoeveel stappen zou een lineaire zoektocht naar een getal in beslag nemen? Antwoord in termen van n, zowel in het beste als in het slechtste geval.

ChatGPT

Bij een lineaire zoektocht wordt elk element in de reeks één voor één gecontroleerd totdat het gezochte getal wordt gevonden of totdat alle elementen zijn doorlopen.

In het beste geval zou het getal dat je zoekt, zich aan het begin van de reeks bevinden, dus bij de eerste poging zou het gevonden worden. Dus in het beste geval zou een lineaire zoektocht 1 stap kosten.

In het slechtste geval zou het getal dat je zoekt zich aan het einde van de reeks bevinden of helemaal niet in de reeks voorkomen. In dit geval zou je alle n elementen moeten doorlopen voordat je het gezochte getal vindt of vaststelt dat het niet in de reeks zit. Dus in het slechtste geval zou een lineaire zoektocht n stappen kosten.

Voor de gegeven reeks {2, 4, 6, 8, 10, 12, 13} zou het vinden van het getal 13 in het beste geval 1 stap kosten en in het slechtste geval 7 stappen.

Figuur 1.1: Conversatie met ChatGPT-3.5 waarbij een vraag omtrent tijdscomplexiteit behandeld wordt.

maar geeft geen bronnen voor de gegenereerde teksten. Het debat over de integratie van AI in het onderwijs kent dan ook twee kanten. Enerzijds zijn er verschillende opportuniteiten door het gebruik van AI-gebaseerde leertechnologieën. Het gebruik van AI binnen het onderwijs zou potentieel kunnen leiden tot geïndividualiseerd leren, verbeterde toegang tot onderwijsmiddelen en efficiëntere leermethoden (Tlili et al., 2023).

Aan de andere kant zijn er ook uitdagingen met het gebruik van AI binnen het onderwijs. Er zijn verschillende ethische vraagstukken, privacykwesties en zorgen over de vervanging van menselijke interactie en het verlies van empathie in het leerproces. Te veel vertrouwen in AI binnen het onderwijs kan leiden tot het wegnemen van de menselijke factor en het creëren van ongelijkheden voor leerlingen die niet optimaal voordeel kunnen halen uit technologische vooruitgang (Chiu et al., 2023).

Deze opportuniteiten en uitdagingen weerspiegelen de complexiteit van het integreren van AI in het onderwijs. Het vereist niet alleen een grondige evaluatie van de voordelen en uitdagingen, maar ook een evenwichtige aanpak om de kracht van AI te benutten terwijl menselijke interactie behouden blijft. Verder zijn veel studenten en docenten nog niet op de hoogte van de capaciteiten van de nieuwe LLM's waardoor het moeilijk wordt om hiermee om te gaan. Het onderwijssysteem staat voor de uitdaging om AI op een verantwoorde manier te omarmen, waarbij zowel innovatie als menselijke waarden in het leerproces worden gewaarborgd.

1.2 Probleemstelling

Het potentieel van AI-tools om het leerproces te verrijken is al eerder onderzocht. Chiu et al. (2023) noemen het voeren van mens-machineconversaties, het toekennen van gepersonaliseerde taken en het ondersteunen van het maken van educatieve beslissingen als enkele voorbeelden van de mogelijke toegevoegde waarde die AI kan brengen binnen het onderwijs. Het effectieve gebruik ervan vereist echter een diepgaand begrip van verschillende kwesties. De Katholieke Universiteit Leuven heeft onder andere al een aantal richtlijnen en waarschuwingen gepubliceerd in verband met het gebruik van generatieve AI zoals ChatGPT (Katholieke Universiteit Leuven, 2024b). Om de mogelijke opportuniteiten en uitdagingen beter in kaart te brengen wordt in deze masterproef gefocust op drie probleemstellingen.

1.2.1 Hoe kan de onderwijssector voordeel halen uit het gebruik van AI-tools?

Het probleem dat de onderwijssector tegenkomt ligt vaak in de beperkte mogelijkheden om gepersonaliseerd onderwijs op grote schaal te bieden. Dit beïnvloedt zowel docenten als studenten. Voor docenten betekent dit vaak een enorme werkdruk en uitdagingen bij het individualiseren van onderwijs voor verschillende leerstijlen en niveaus binnen één groep. Voor studenten kan dit resulteren in minder betrokkenheid, een gebrek aan uitdaging of ondersteuning die is afgestemd op hun specifieke behoeften.

De oorzaak van dit probleem ligt deels in de traditionele onderwijsmethoden die niet altijd in staat zijn om rekening te houden met de diversiteit aan leerstijlen en individuele behoeften van studenten (Rowe, 2006). Daarnaast kan de beperkte beschikbaarheid van middelen, zoals tijd en gekwalificeerd personeel, de mogelijkheid om elke student de aandacht te geven die hij verdient, beperken (De Morgen, 2023).

Het gevolg van deze beperkingen is dat veel studenten niet het maximale uit hun onderwijservaring halen. Sommigen raken achterop, terwijl anderen zich mogelijk vervelen of niet uitgedaagd voelen. Dit kan resulteren in een gebrek aan motivatie, lagere academische prestaties en zelfs een verminderde interesse in leren op lange termijn.

Het gebruik van AI-tools biedt een potentieel antwoord op dit probleem. Deze tools hebben de mogelijkheid om data te analyseren en patronen te identificeren die kunnen helpen bij het personaliseren van het onderwijs. Door adaptieve leermiddelen en AI-ondersteunde leersystemen kunnen docenten meer gerichte ondersteuning bieden aan individuele studenten, waardoor ze beter kunnen inspelen op diverse leerstijlen en behoeften. Dit kan resulteren in een verbeterde betrokkenheid, hogere prestaties en een meer inclusieve leeromgeving (Kasneci et al., 2023).

1.2.2 Wat zijn de uitdagingen verbonden met het inzetten van AI binnen het onderwijs?

Het inzetten van AI binnen het onderwijs is een complexe onderneming die gepaard gaat met diverse obstakels en vraagstukken. Het verkennen van deze uitdagingen vormt een belangrijk aspect van de literatuurstudie. Deze literatuurstudie zal later dienen als een basis waarop een prototype zal worden gebaseerd om deze uitdagingen aan te pakken. Hierbij kan gedacht worden aan de pedagogische benaderingen, de aanpassing aan diverse leerstijlen en de overkoepelende

onderwijsdoelen. Ook het black-boxprincipe krijgt hier aandacht. Dit is een probleem bij de huidige AI-modellen doordat ze zo groot en complex geworden zijn dat zelfs de ontwikkelaars vaak moeite hebben om de werking uit te leggen (Wadden, 2021). Vandaar de naam, black-box. Aan de ene kant wordt er een input gegeven en aan de andere kant volgt er een bepaalde output. Wat daar tussenin gebeurd is echter bijzonder moeilijk om uit te leggen.

Een ander vraagstuk is de effectieve implementatie van AI zonder de menselijke interactie en het kritische denkvermogen van leerlingen te verwaarlozen. Het evenwicht tussen technologie en de waardevolle rol van docenten in het leerproces is van groot belang. Het is cruciaal om AI niet te zien als vervanging, maar eerder als een ondersteunend instrument om het onderwijs te verrijken.

Een laatste belangrijk vraag is de toegankelijkheid en gelijkheid in het gebruik van AI-tools. Het is van belang dat deze technologische middelen breed toegankelijk zijn voor alle studenten, ongeacht hun achtergrond, om te voorkomen dat technologische vooruitgang de bestaande ongelijkheden in het onderwijs vergroot.

1.2.3 Hoe kunnen de oplossingen voor deze uitdagingen geïntegreerd worden?

Nadat de mogelijke toepassingen van AI binnen het onderwijs en de daarbij gepaard gaande uitdagingen zijn vastgesteld, komt het aspect van implementatie naar voren. Het ontwikkelen van een effectieve implementatiestrategie voor AI in het onderwijs is van essentieel belang. Om deze strategie te onderzoeken en demonstreren wordt een prototype ontwikkelt dat geselecteerde technologieën implementeert.

Dit prototype wordt ontworpen als een hulpmiddel dat meerdere doelen dient. Enerzijds zal het studenten bijstaan bij het maken van oefeningen, waarbij AI-technologie wordt ingezet om adaptieve leermiddelen te bieden, afgestemd op hun behoeften. Anderzijds zal de applicatie tools en inzichten verschaffen aan docenten om het leerproces te begeleiden en aan te passen aan de verschillende leerbehoeften van hun studenten.

1.3 Doelstellingen

De hoofddoelstelling van het project is het ontwikkelen van een prototype dat concreet demonstreert hoe artificiële intelligentie effectief geïntegreerd kan worden in het onderwijs. Het prototype moet aantonen welke AI-tools nuttig kunnen zijn binnen een educatieve context en hoe huidige uitdagingen aangepakt kunnen worden. Dit doel wordt opgedeeld in drie deeldoelstellingen.

1.3.1 Bepalen van nuttige AI-tools en hun uitdagingen binnen het onderwijs

In deze doelstelling staat de vraag "Hoe kan de onderwijssector voordeel halen uit het gebruik van AI-tools?"centraal. Hiervoor wordt eerst een literatuurstudie gedaan waarin huidige toepassingen en hun uitdagingen besproken worden. Vervolgens wordt er geëxperimenteerd door enkele oefeningen op te lossen met LLM's en de antwoorden te bespreken met docenten. Concreet moet aan de volgende eisen voldaan worden:

- Overzicht tonen van huidige AI-gebaseerde educatieve tools.
- Identificeren van andere mogelijke toepassingen van AI-gebaseerde educatieve tools.
- Identificeren van uitdagingen met nieuwe AI-gebaseerde educatieve tools.
- Experimenteren door oefeningen met LLM's op te lossen en docenten te interviewen om de antwoorden te bespreken.

1.3.2 Ontwikkelen van een prototype dat studenten ondersteunt bij het maken van oefeningen

Deze doelstelling gaat over het eerste deel van het prototype. Dit deel is gericht op studenten en moet hen ondersteunen bij het maken van oefeningen. Het ontwerp van dit prototype wordt gebaseerd op de conclusies die getrokken kunnen worden uit de antwoorden op de eerste doelstelling. Hoofdstuk 3 heeft hiervoor de volgende eisen vastgesteld:

- Toegang geven tot een LLM op een verantwoorde manier.
- Aanbieden van een FAQ die dynamisch aangepast wordt aan de hand van vragen van de groep studenten.
- Aanbevelen van oefeningen aan de student op basis van eerdere struikelpunten.

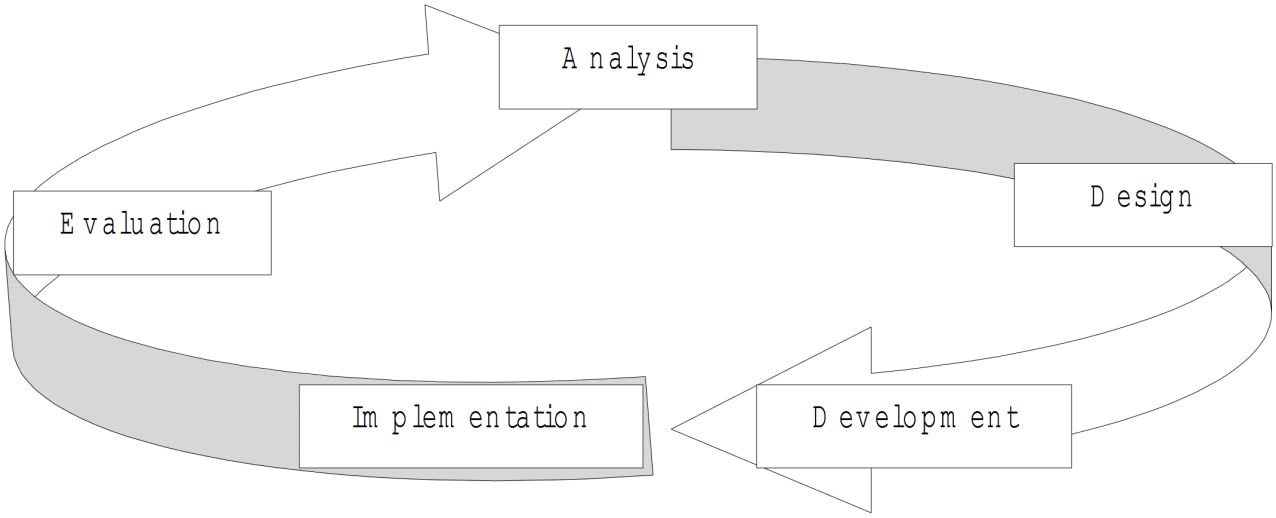
1.3.3 Ontwikkelen van een prototype dat docenten ondersteunt bij het aanleren van oefeningen

In deze doelstelling wordt gekeken naar het tweede deel van het prototype. Hierin wordt gericht op het ondersteunen van docenten bij het aanleren van oefeningen. Er wordt ook gekeken naar hoe docenten meer inzicht kunnen krijgen in de prestaties van de leerlingen doorheen de lesperiode. Hoofdstuk 3 stelt vast dat dit deel van de applicatie de volgende functies moet bevatten:

- Tonen van veelgestelde vragen aan de docent en een antwoord voorstellen om toe te voegen aan de FAQ.
- Controle geven over het type LLM dat studenten kunnen gebruiken en specifieke instructies aan dit LLM geven.
- Suggesties tonen die aangeven waar studenten moeilijkheden ondervinden. Deze suggesties moeten gestaafd zijn door data om de applicatie zo transparant mogelijk te laten werken.
- Creëren van visualisaties die docenten ondersteunen bij het verder ontwikkelen van hun lesmateriaal door moeilijkheden aan te duiden.

1.4 Methode

Het werkproces is gestructureerd in verschillende werkpakketten (WP). Elk werkpakket is zorgvuldig samengesteld met specifieke taken en doelen. Deze vormen allemaal een essentieel onderdeel van het traject naar het behalen van een succesvol resultaat. Voor elk werkpakket zal gesteund worden op het ADDIE proces (Peterson, 2003). Dit proces is ook te zien in Figuur



Figuur 1.2: Het ADDIE proces zoals toegepast in de werkpakketten (Peterson, 2003).

1.2. Dit proces start met een grondige analyse van het probleem. Dit vindt plaats in de eerdere probleemstelling (zie 1.2). Vervolgens worden concrete eisen gesteld waaraan voldaan moet worden. Dit vindt plaats in sectie 1.3. Vervolgens vinden de ontwikkeling- en implementatiestap plaats. Hierin wordt een oplossing voor het probleem ontwikkelt dat voldoet aan de eisen, om dit vervolgens te implementeren. Ten slotte wordt er ook een evaluatie uitgevoerd. Hierin wordt nagegaan of de ontwikkelde oplossing aan alle eisen voldoet. Dit vindt plaats in hoofdstuk 6. Deze laatste stap, zoals beschreven in werkpakket 3, zal een kritische reflectie bieden op het geïmplementeerde prototype en toekomstige onderzoeksrichtingen bespreken.

1.4.1 WP1: Identificeren van educatieve toepassingen

Dit werkpakket heeft als doelstelling het identificeren van educatieve toepassingen binnen onderwijsomgevingen die potentieel kunnen worden verbeterd door de integratie van artificiële intelligentie. Het werkpakket zal zich richten op het verzamelen, analyseren en evalueren van bestaande educatieve toepassingen om de mogelijkheden van AI binnen deze contexten te begrijpen en aanbevelingen te formuleren voor verdere ontwikkeling en implementatie.

Het werkpakket zal zich ook richten op het onderzoeken en begrijpen van uitdagingen omtrent AI binnen het onderwijs om inzicht te krijgen in de beperkingen, ethische dilemma's, technologische barrières en sociale implicaties van AI-gebruik. Het identificeren van deze obstakels is van cruciaal belang om een gefundeerde aanpak te ontwikkelen voor het effectief en verantwoord inzetten van AI in verschillende contexten. Het resultaat van dit werkpakket moet de eisen van sectie 1.3.1 beantwoorden.

1.4.2 WP2: Implementatie en evaluatie van het educatieve prototype

Dit werkpakket is gericht op het ontwikkelen van een prototype dat specifieke problemen aanpakt die geïdentificeerd zijn tijdens de eerdere fasen van het onderzoek. Dit prototype moet beantwoorden aan de doelstellingen zoals beschreven in secties 1.3.2 en 1.3.3. Dit betekent dat het prototype een bijdrage moet leveren aan het leerproces van de student door ondersteuning te bieden bij het zelfstandig verwerken van de oefeningen en door ondersteuning te bieden aan

de docent bij het aanleren van de oefeningen. Hiervoor zullen aan de hand van de Windows Presentation Foundation (WPF) en C# twee applicaties uitgewerkt worden, één voor de studenten en één voor de docenten. Achterliggend zullen deze applicaties communiceren door een gemeenschappelijke Pythonserver.

Na het implementeren van het prototype voor zowel de studenten als de docenten zal een kritische reflectie uitgevoerd worden. Deze reflectie zal later gebruikt worden om een besluit te vormen en toekomstig onderzoek voor te stellen.

1.4.3 WP3: Schrijven van de thesis

Dit werkpakket richt zich op het schrijfproces van de masterthesis, met als doel een goed gestructureerd document te produceren dat alle onderzoeksvragen beantwoordt. Het begint met een literatuuronderzoek rond AI-toepassingen in het onderwijs en de uitdagingen hierrond. Het schrijfproces omvat bespreking van onderzoeksvragen, methodologie, bevindingen en conclusies met strikte naleving van academische normen.

Het uiteindelijke succes van dit werkpakket zal worden gemeten aan de hand van de kwaliteit van de opgeleverde thesis, waarin niet alleen de onderzoeksvragen beantwoord en relevante bevindingen gepresenteerd worden, maar waarin ook aan het vakgebied en de academische gemeenschap wordt bijgedragen.

1.5 Vooruitblik

In hoofdstuk 2 wordt een grondige bronnenstudie uitgevoerd die de verschillende toepassingen en uitdagingen met betrekking tot artificiële intelligentie binnen het onderwijs moet bepalen.

Hoofdstuk 3 zal op basis van de bronnenstudie LLM's verder exploreren en focussen op het oplossen van oefeningen met LLM's. De bevindingen worden gebruikt voor de ontwikkeling van een concept voor het uiteindelijke prototype.

Vervolgens worden in hoofdstuk 4 de ontwikkelde prototypes voor docenten en studenten getoond. Vervolgens wordt de architectuur van het systeem besproken. Dit omvat de werking van elke individuele component en de communicatie tussen de verschillende componenten.

Hoofdstuk 5 bevat een kritische reflectie van het prototype. Deze bestaat eerst uit een overzicht van de technologische beperkingen gevolgd door mogelijke uitbreidingen op het prototype. Verder wordt de privacy van studenten die het prototype gebruiken besproken, om vervolgens af te sluiten met de opzet van een mogelijke user study.

Uiteindelijk vormt hoofdstuk 6 een besluit waarin teruggekeken wordt op de masterproef en de belangrijkste conclusies herhaald worden.

Hoofdstuk 2

Toepassingen en uitdagingen van AI in het onderwijs

In dit hoofdstuk worden eerst de verschillende toepassingen besproken van artificiële intelligentie in het onderwijs. Er worden enkele concrete voorbeelden gegeven die aantonen hoe AI al eerder is toegepast en eventuele opmerkingen bij deze toepassingen worden ook besproken. Daarna worden de uitdagingen besproken die gepaard gaan met het gebruik van AI. Hier wordt de nadruk gelegd op LLM's aangezien deze aan de basis liggen van het verdere prototype.

2.1 Toepassingen

Zhang and Aslan (2021) delen AI-toepassingen binnen het onderwijs op in zes verschillende types. In deze sectie worden eerst machine learning en Personalized learning systems besproken. Dit zijn de algemene concepten waar de andere toepassingen zich binnen specialiseren.

2.1.1 Machine learning

Machine learning (ML) kan zeer breed ingezet worden voor het analyseren van zowel docenten als studenten. Zo hebben Wei et al. (2018) machine learning gebruikt om te onderzoeken hoe de studiestijl van studenten verandert doorheen hun academische periode. Het systeem in dit onderzoek gebruikt de vragen uit het Felder-Silverman model voor leerstijlen (Felder and Silverman, 2002). De studenten beantwoorden de vragen, waarna deze door een support vector machine (SVM) geanalyseerd worden. Deze inzichten kunnen docenten helpen om hun onderwijsmethoden aan te passen aan de veranderende behoeften van hun studenten. Bovendien kan machine learning ook worden gebruikt om de prestaties en effectiviteit van docenten te analyseren. Door het gedrag en de interacties van docenten in de klas te observeren, kunnen machine learning algoritmen waardevolle feedback geven over hun onderwijsstijl (Kushik et al., 2020). Dit kan docenten helpen om hun methodes te verbeteren en uiteindelijk tot betere leerresultaten voor hun studenten leiden.

2.1.2 Personalized learning systems

Personalized learning systems (PLS) vormen een innovatieve benadering binnen e-learning platformen, die als doel hebben om zelfstandig leren aanzienlijk te vereenvoudigen en te verbeteren.

Een voorbeeld van een PLS is het systeem van Köse (2017). Dit systeem is specifiek ontworpen voor mobiele telefoons en maakt naast AI ook gebruik van augmented reality (AR) om de leerervaring te personaliseren. AI wordt ingezet om de juiste studiematerialen voor studenten aan te bevelen op basis van diverse parameters die betrekking hebben op hun studiegedrag. Hierdoor kunnen studenten effectiever leren en zich richten op de inhoud die het meest relevant is voor hun behoeften en interesses.

Een ander interessant voorbeeld is het systeem van Walkington and Bernacki (2019), waarbij AI wordt ingezet om het leren van algebra te vereenvoudigen. In dit geval wordt artificiële intelligentie gebruikt om oefeningen te koppelen aan de persoonlijke interesses van elke student. Dit maakt het leerproces niet alleen persoonlijker, maar ook boeiender en relevanter voor de student, wat de motivatie en betrokkenheid bij het leerproces aanzienlijk kan verhogen.

2.1.3 Chatbots

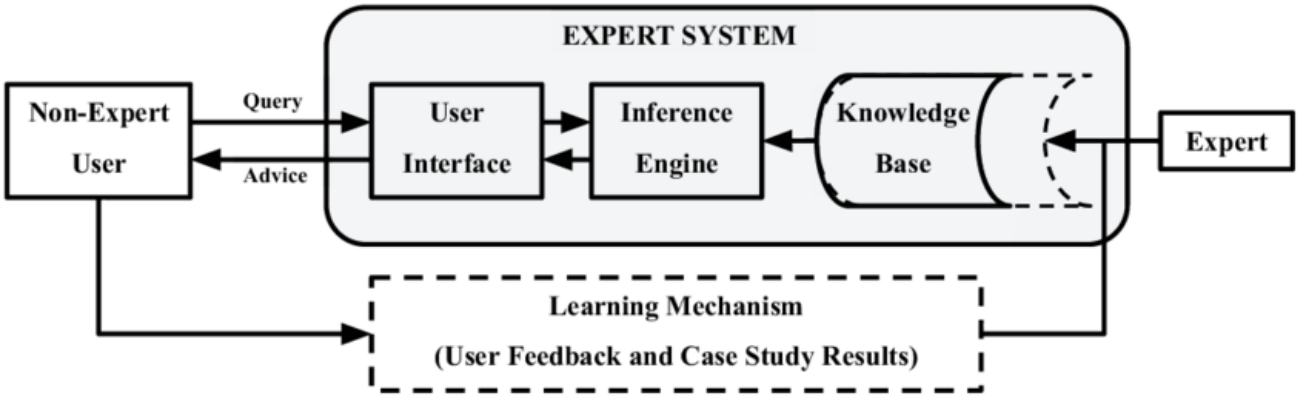
Chatbots zijn geen nieuwe technologie. Een van de eerste chatbots, ELIZA, werd in 1966 al voorgesteld (Weizenbaum, 1966). Recent is er echter te spreken van een revolutie door de ontwikkeling van Large Language Models (LLM's), zoals bijvoorbeeld ChatGPT (OpenAI, 2023b). Dit zijn generatieve AI-modellen die getraind zijn op grote hoeveelheden data waardoor ze een goed begrip van taal krijgen. Hierdoor kunnen ze een gesprek voeren zoals een persoon dat kan. De modellen zijn in staat om tekst van een persoon te interpreteren en beantwoorden met natuurlijke taal. LLM's zijn ook in staat om gesprekken te voeren over diverse onderwerpen zonder daar specifiek getraind voor te zijn. LLM's worden verder besproken in sectie 3.1. De chatbots in de volgende voorbeelden zijn niet gebaseerd op LLM's. Hier kan dan ook opgemerkt worden dat ze allemaal een zeer specifiek doel hebben.

De toepassingen van chatbots zijn talrijk. Een groot aantal chatbots wordt ingezet binnen klantendiensten (Johannsen et al., 2018). Door interacties met de chatbot kunnen klanten hun probleem mogelijks zelf al oplossen en anders kan de chatbot hen doorverbinden met de juiste persoon. Ook binnen de horeca zijn chatbots al ingezet om klanten verder te helpen. Zo heeft restaurantketen Domino's Pizza een chatbot gelanceerd die de bestellingen van klanten kan opnemen via Facebook Messenger (Gilliland, 2016).

Ook binnen het onderwijs is er geen tekort aan toepassingen voor chatbots. Fryer et al. (2017) onderzochten het gebruik van een chatbot om Engels te leren in een Japanse Universiteit. De studenten werden onderverdeeld in twee groepen en voltooiden oefeningen met beurtelings de hulp van een persoon en de chatbot. Opvallend was hier dat de interesse in het vak daalde bij het gebruik van de chatbot, terwijl dit niet het geval was bij een menselijke partner. Dit toont aan dat de menselijke kant in het onderwijs toch ook belangrijk is. Dit wordt verder besproken in sectie 2.2.7.

2.1.4 Expert systems en intelligent tutors

Een expert system is een manier om grote hoeveelheden domein-specifieke data bereikbaar te maken voor personen die weinig kennis hebben over dit domein (Saibene et al., 2021). Het systeem kan beslissingen nemen zoals een domeinexpert op basis van verstrekte gegevens door de gebruiker. De werking van een expert system wordt gegeven door Figuur 2.1. Een praktisch voorbeeld wordt gegeven door Chen (2011). In dit onderzoek is een hartslagsensor gekoppeld



Figuur 2.1: Werking van een expert system. De kennis van een expert is beschikbaar voor non-experts door een kennisbank en vastgelegde regels in de Inference Engine (Ozden et al., 2016).

aan een expert system. Dit expert system analyseert de data van de patiënt en zal, zoals een dokter, vaststellen of het hartritme onregelmatig is en een diagnose uitvoeren.

Hwang et al. (2020) tonen dat een expert system ook binnen het onderwijs toegepast kan worden. Zij ontwikkelden een expert system dat bij studenten zowel de emotionele en cognitieve factoren in rekening neemt om vervolgens een adaptieve leeromgeving te creëren. Hun onderzoek toont aan dat studenten die traditioneel slechter presteren meer taken voltooiden met het expert system dan de controlegroep die op een conventionele manier studeerde.

Intelligent tutors zijn een doorontwikkeling van expert systems die in diverse vormen kunnen voorkomen, waardoor ze zich kunnen aanpassen aan verschillende leerbehoeften en -situaties (Passey et al., 2018). Deze veelzijdige digital agent kan dienen als een docent, mentor, instructeur, coach, studiegenoot en zelfs als een leerling (Tärning et al., 2019).

Een interessant voorbeeld van het gebruik van een intelligent tutor wordt gegeven door Tärning et al. (2019), waarbij de digital agent de rol van een leerling op zich nam. Dit experiment vereiste dat de echte studenten de leerstof aan de digital agent overbrachten, wat een unieke dynamiek creëerde in het leerproces. De bevindingen van deze studie tonen aan dat het gebruik van een intelligent tutor aanzienlijke voordelen kan bieden aan studenten. Enerzijds wordt de interactie met een digitale leerling als een waardevolle oefening beschouwd, waarbij studenten hun begrip van de leerstof verder kunnen verdiepen door deze aan anderen uit te leggen. Anderzijds kunnen intelligent tutors studenten helpen bij het ontwikkelen van hun vermogen om zelfstandig leerstof te verwerken, aangezien de digital agents op maat gemaakte begeleiding en ondersteuning kunnen bieden.

De onderzoeken van Hwang et al. (2020) en Tärning et al. (2019) tonen aan dat een aangepaste studiemethode een voordeel kan zijn voor studenten. Het prototype implementeert een recommender system (zie sectie 4.1.2) om de studenten ook te ondersteunen door een vorm van adaptief leren aan te bieden, al gaat het in het prototype niet specifiek om een expert system.

2.1.5 Visualizations en virtual learning environments

Dankzij de toenemende populariteit van virtual reality (VR) wordt er ook onderzoek gedaan naar hoe dit kan toegepast worden binnen het onderwijs om een virtual learning environment (VLE) te creëren. Binnen deze VLE's kan dan ook AI geïntegreerd worden om de studenten en

docenten verder bij te staan. Zo hebben Ijaz et al. (2017) een historische stad nagebouwd waarin studenten in virtual reality (VR) kunnen rondwandelen en interageren met de bewoners van de stad. Deze bewoners worden aangestuurd door AI en kunnen hierdoor reageren op hun omgeving en andere bewoners. De door AI bestuurde bewoners kunnen de studenten ook meer uitleg geven over hun rol binnen de stad en de samenleving.

Een ander voorbeeld, eveneens mee ontwikkeld door onder andere EDM, is de Virtual Nuclear Simulator (ViNuS) (Universiteit Hasselt, 2024b). ViNuS is een VR-simulatie om de overgang tussen theorie en praktijk te vergemakkelijken voor nucleaire toepassingen. Een voorbeeld van een toepassing is het werken met bronnen van nucleaire energie, maar dit is eveneens toepasbaar binnen de medische wereld. Hierbij is een toepassing bijvoorbeeld het uitvoeren van een CT-scan (Universitair Ziekenhuis Leuven, 2024).

2.2 Uitdagingen

Hoewel er duidelijk potentieel zit in bovenstaande toepassingen voor het gebruik van artificiële intelligentie binnen het onderwijs, zijn er ook nog enkele struikelpunten. Tlili et al. (2023) hebben verschillende scenario's aangeduid waarin AI en LLM's nog tekort komen om wijde inzet mogelijk te maken. De belangrijkste tekortkomingen voor gebruik binnen het onderwijs worden hierna besproken en vergeleken met bevindingen uit bijkomende bronnen. Ook de gids over verantwoord gebruik van generatieve AI, gepubliceerd door de Katholieke Universiteit Leuven (2024b), wordt hierin besproken. Ten slotte wordt er ook gefocust op de interactie tussen studenten en docenten en hoe deze beïnvloed kan worden door AI.

2.2.1 Impact op evaluaties

Vooral met de opkomst van chatbots zoals ChatGPT worden er veel vragen gesteld over hoe dit binnen een academische context correct gebruikt kan worden. Vooral bij het evalueren van teksten is het moeilijk voor docenten om te detecteren of deze door een persoon of een AI-model geschreven zijn. Gao et al. (2022) vonden dat uit een groep AI-gegenereerde abstracten 68% correct geïdentificeerd werd door de testpersonen. Deze testgroep identificeerde ook 14% van de door personen geschreven abstracten als AI-gegenereerd.

Omdat het voor een persoon dus moeilijk is correct vast te stellen of een tekst AI-gegenereerd is, hebben verschillende onderzoeken geprobeerd om zogenaamde *GPT output detectors* te bouwen. Dit zijn softwareprogramma's die voor een gegeven tekst een zekerheidspercentage geven dat de tekst door AI of een persoon is geschreven (Jawahar et al., 2020). Tlili et al. (2023) hebben dit getest door een tekst volledig door ChatGPT te laten schrijven en deze vervolgens door een GPT output detector te laten nakijken. Bij dit specifieke geval werd de tekst als computergegenereerd bestempeld met een zekerheid van ongeveer 99%. Opvallend is echter het resultaat nadat de onderzoekers een enkel woord toevoegen aan de tekst. De tekst wordt nu met ongeveer 75% zekerheid bestempeld als geschreven door een persoon.

Ook Mitchell (2023) rapporteert hoe deze detectors onbetrouwbaar zijn. Er werd vastgesteld dat GPTZero[1], een van de vele tools om AI-geschreven teksten te detecteren, de grondwet van de Verenigde Staten van Amerika als AI-geschreven detecteerde. Dat deze tools onbetrouwbaar zijn

[1] `https://www.gptzero.me/`

heeft ook de KU Leuven vastgesteld. Zij geven het volgende advies aan hun docenten in verband met het detecteren van AI-gegenereerde teksten:

> "Beschouw de AI-scores van AI-detectoren als louter indicatief. We manen aan tot voorzichtigheid gezien AI-detectoren ook vals negatieven en vals positieven kunnen genereren." - Katholieke Universiteit Leuven (2024a)

2.2.2 Betrouwbaarheid en consistentie van antwoorden

Een ander veelbesproken punt is de betrouwbaarheid van de antwoorden die ChatGPT geeft. Wagner and Ertl-Wagner (2023) hebben onderzocht hoe ChatGPT gebruikt kan worden om vragen te beantwoorden in verband met radiologie in een ziekenhuis. In dit onderzoek zijn er 88 vragen gesteld aan ChatGPT, waarvan er 59 (67%) juist beantwoord werden en 29 (33%) fout beantwoord werden. Vervolgens werden er ook referenties gevraagd om de antwoorden te onderbouwen. ChatGPT gaf de onderzoekers 343 referenties, echter bleken er slechts 124 (36.2%) echt te bestaan met 219 (63.8%) neppe referenties. Ook stellen de onderzoekers dat van de 124 bestaande referenties slechts 47 (37.9%) ook voldoende informatie gaven over de specifieke situatie.

Figuur 2.2 toont een voorbeeld van een verkeerd antwoord. Er wordt aan ChatGPT-3.5 gevraagd om de tijdscomplexiteit van binair zoeken te bepalen. De redenering achter het slechtste geval is correct, maar de redenering voor het beste geval niet. In het beste geval zal het middelste element het gezochte element zijn waardoor er slechts 1 stap nodig is. Deze figuur toont ook een fenomeen dat bij LLM's bekend staat als *confidently incorrect*, oftewel, vol vertrouwen een fout antwoord geven. Een student kan snel geloven dat het LLM een correct antwoord gegeven heeft omdat het niet twijfelt. Het LLM is zeker dat het antwoord correct is.

Van een docent wordt ook verwacht dat wanneer verschillende studenten dezelfde vraag stellen, de docent hen hetzelfde antwoord zal geven. Bij ChatGPT is dit echter niet het geval. In het onderzoek van Tlili et al. (2023) stelden drie docenten elk identiek dezelfde vraag. De antwoorden waren echter compleet verschillend. In de context van onderwijs is dit natuurlijk een groot probleem aangezien leerlingen verschillende informatie krijgen en de kansen dus niet gelijk zijn bij, bijvoorbeeld, een examen.

2.2.3 Naïviteit en structuur van de gecreëerde opdrachten

Een andere toepassing van AI binnen onderwijs kan het genereren van opdrachten en testen voor de docent zijn. Hierdoor wordt de werkdruk van de docent mogelijk verlicht. Zo vroegen Tlili et al. (2023) aan ChatGPT om een multiple-choice test op te stellen. Er wordt opgemerkt dat de verkeerde antwoorden steeds zeer opmerkelijk waren en makkelijk aan te duiden zonder veel kennis over het onderwerp van de test. Verder werd het foute antwoord ook steeds als laatste optie gegeven. Als voorbeeld werd ChatGPT gevraagd om een meerkeuzevraag op te stellen over educatieve technologie. De vraag die bedacht werd is: "Welke van de volgende is GEEN educatieve technologie?". De gegeven keuzes waren: "leermanagementsysteem", "virtual reality headset", "smartboard" en "pizzaoven".

Het is ook belangrijk voor studenten om een duidelijke opgave te ontvangen. De layout is best consistent en toont duidelijk wat er verwacht wordt van de student. Wanneer ChatGPT ingezet wordt om opdrachten te maken valt het echter op dat de layout van de output willekeurig is.

Anonymous

Gegeven is de volgende reeks: {2, 4, 6, 8, 10, 12, 13}. Hoeveel stappen zou een binaire zoektocht naar een getal in beslag nemen? Antwoord in termen van n, zowel in het beste als in het slechtste geval.

ChatGPT

Bij een binaire zoektocht wordt de reeks opgesplitst in steeds kleinere delen door telkens het middelste element te vergelijken met het gezochte getal. Het aantal stappen dat nodig is, hangt af van de lengte van de reeks en de positie van het gezochte getal.

In het beste geval zou het gezochte getal meteen in het midden van de reeks worden gevonden. In dit geval zou het aantal stappen log2(n) zijn, waarbij n het aantal elementen in de reeks is. Dus, bij een reeks van 7 elementen zoals in jouw geval, log2(7) is ongeveer 2,81, wat afgerond naar boven 3 stappen zou betekenen.

In het slechtste geval zou het gezochte getal aan de uiterste randen van de reeks liggen, waardoor je alle stappen van het splitsen zou moeten doorlopen voordat je het vindt. Dit zou ook log2(n) stappen zijn, maar afgerond naar boven. Dus in dit geval zou het ook ongeveer 3 stappen kosten.

Dus, in het beste en slechtste geval zou een binaire zoektocht in deze reeks ongeveer 3 stappen kosten.

Figuur 2.2: Voorbeeld van een verkeerd antwoord van ChatGPT-3.5. In het beste geval is het aantal stappen 1.

Tlili et al. (2023) merken op dat er vaak veel verschillen waren tussen gegenereerde testen die extra inspanning vragen van de docent om alles te herwerken naar een bruikbaar formaat. Zo zal ChatGPT de meerkeuze-antwoorden soms op één regel tekst schrijven en soms elk op een aparte regel. Bovendien wordt soms ook het correcte antwoord gegeven in de output, maar andere keren ook niet.

2.2.4 Het volledige potentieel benutten

Niet elke student heeft dezelfde talenkennis en voor veel studenten is Engels, wat voor veel chatbots de enige voertaal is, geen moedertaal. Van docenten wordt verwacht dat zij geen onderscheid maken tussen studenten met verschillende taalniveaus. Uit het onderzoek van Tlili et al. (2023) blijkt echter dat de manier waarop vragen gesteld worden aan ChatGPT zeer belangrijk is. Wanneer een vraag iets anders geformuleerd wordt kan het antwoord totaal anders worden. Hierdoor is het dus ook belangrijk dat de student een grondige talenkennis heeft om de vragen zo goed en gedetailleerd mogelijk op te stellen.

Ook wanneer de student een grondige talenkennis heeft, is het nog steeds belangrijk hoe de vraag gesteld wordt. Net zoals bij een zoekmachine kunnen de resultaten zeer verschillend zijn wanneer de vraag iets anders verwoord wordt. Wu and Hu (2023) tonen dat de kwaliteit van een antwoord sterk afhankelijk is van de gegeven context. Bij het gebruik van een LLM om teksten te vertalen kregen zij betere resultaten als ze het LLM meer context gaven over de situatie. Deze technieken, bekent als *prompt engineering*, worden verder besproken in sectie 3.2.2.

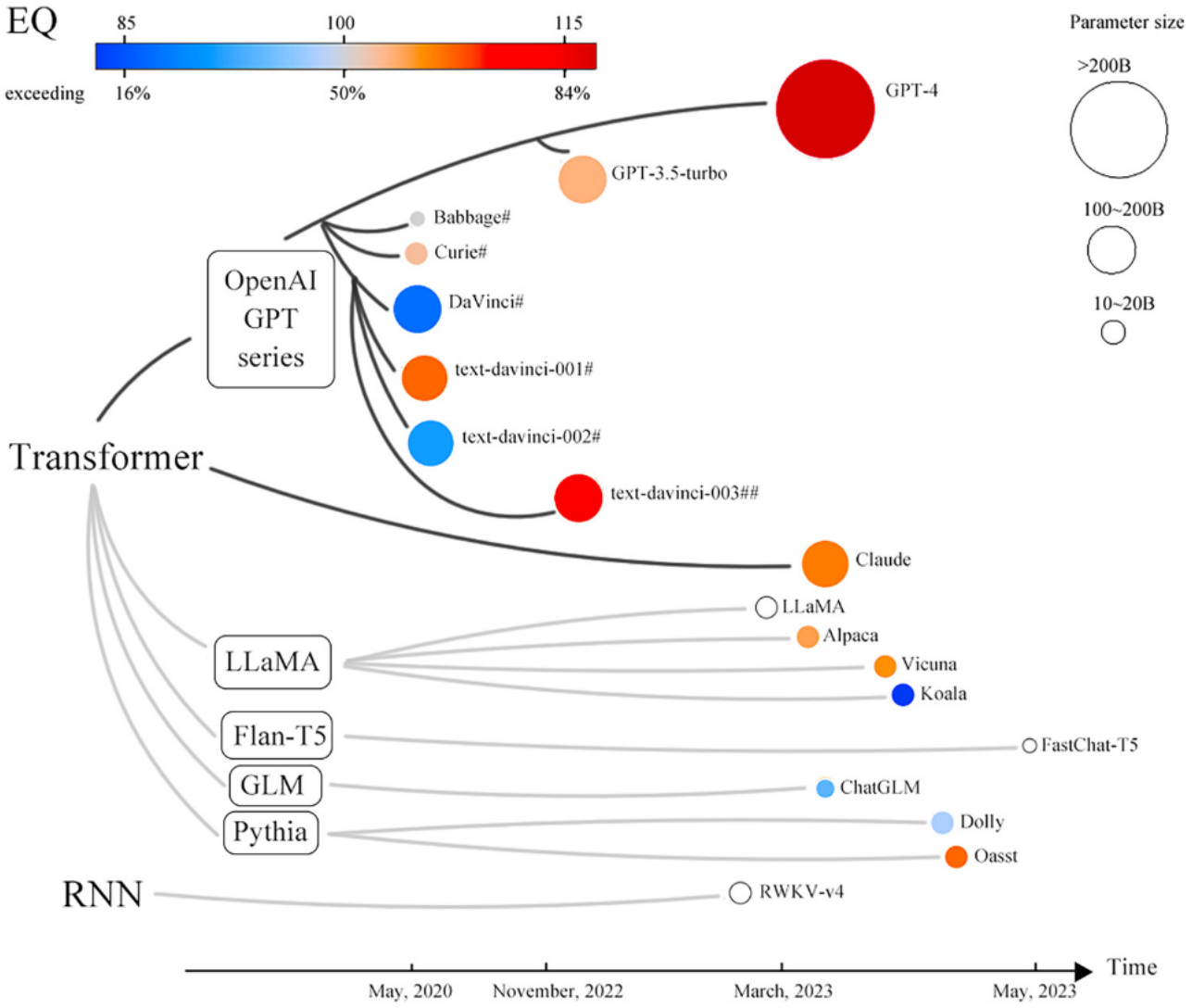
2.2.5 Gebrek aan reflectie en emotie

Vaak vraagt een docent aan het einde van een semester of academiejaar om feedback te geven zodat ze hun methode of cursus kunnen verbeteren voor volgende academiejaren. ChatGPT kan echter geen feedback onthouden. Hiervoor zou het model opnieuw getraind moeten worden. Ook het tonen van emotie is voor ChatGPT moeilijk. Tlili et al. (2023) stellen dan ook voor dat toekomstig onderzoek kan focussen op het meer menselijk maken van chatbots, zowel op het gebied van emoties tonen alsook door te kunnen reflecteren. Nieuwe LLM's scoren al wel beter op emotionele testen (Wang et al., 2023). Dit is vooral te danken aan het vergroten van de modellen. Doordat ze meer parameters kunnen trainen kunnen de LLM's ook meer vaardigheden ontwikkelen. Figuur 2.3 toont de Emotionele Quotiënt voor enkele LLM's.

De vraag is echter of emotie een grote noodzaak is voor LLM's in het onderwijs. Dit hangt wellicht ook af van de taak van het LLM. In sectie 2.1.3 werd al vastgesteld dat studenten die een taal leren sneller interesse verliezen wanneer ze oefenen met een chatbot. Hier zou meer emotie mogelijk kunnen bijdragen aan verbetering. Aan de andere kant is het misschien een voordeel als een LLM zich voordoet als een machine en niet als persoon. Studenten gaan een machine mogelijk niet blindelings vertrouwen als deze zich niet voordoet als een persoon (Glikson and Woolley, 2020).

2.2.6 Privacy en intellectuele eigendom

Wanneer studenten vrij kunnen spreken met een chatbot zoals ChatGPT is privacy ook erg belangrijk. Voor veel studenten is een docent ook een vertrouwenspersoon waar ze met problemen terecht kunnen. Het is dan ook niet ondenkbaar dat studenten eenzelfde soort relatie met de



Figuur 2.3: Emotionele Quotiënt (EQ) van verschillende LLM's (Wang et al., 2023).

chatbot verwachten en hun problemen hieraan uitleggen. Het probleem is hier dat de gesprekken kunnen bekeken worden door medewerkers van het bedrijf achter de chatbot en zo dus gevoelige informatie gezien kan worden (OpenAI, 2023a).

In het specifieke geval van ChatGPT werd ook aan de chatbot zelf gevraagd of deze conversaties bewaard en/of gebruikt worden voor trainingsdoelen. Hierbij krijgt de gebruiker het antwoord dat de gesprekken niet bewaard worden en dat het onderliggende LLM niet kan trainen op de data (Tlili et al., 2023). De online documentatie van OpenAI spreekt dit echter tegen en toont dat conversaties wel bewaard kunnen worden (OpenAI, 2023a).

Naast de problemen rond privacy door het trainen op nieuwe data, zijn er ook problemen rond intellectuele eigendom met de data waarop origineel getraind is (European Commision, 2024). Op het moment van schrijven loopt er een rechtszaak tegen OpenAI, aangespannen door de New York Times in december 2023 (Grynbaum and Mac, 2023). Zij claimen dat OpenAI hun modellen heeft getraind op artikels van de New York Times en dat de resulterende chatbots concurreren met de krant waarop ze getraind hebben. De New York Times zou bewijs geleverd hebben dat toont hoe bepaalde vragen het mogelijk maken om volledige artikels als antwoord te krijgen. Deze bewijzen zijn op het moment van schrijven nog niet publiek beschikbaar.

OpenAI heeft gereageerd en zegt dat hun trainingsdata legaal is onder de Amerikaanse *fair use* wetgeving. Deze wet wordt als volgt beschreven:

> "the fair use of a copyrighted work, including such use by reproduction in copies or phonorecords or by any other means specified by that section, for purposes such as criticism, comment, news reporting, teaching (including multiple copies for classroom use), scholarship, or research, is not an infringement of copyright." - U.S. Copyright Office (2024)

Zij stellen dat hun gebruik valt onder research en dus geen inbreuk is op de copyright van de auteurs van de data. De New York Times beargumenteert dit door te stellen dat het product van de research (de chatbots) hier niet onder vallen (Metz, 2024). OpenAI heeft al deals gesloten met andere nieuwsinstellingen voor het gebruik van hun data. Een rechtbank zal nu moeten beslissen of dit inderdaad fair use is of niet. Deze beslissing zal dan ook belangrijk zijn voor de toekomst van Large Language Models.

2.2.7 Interactie tussen docenten en studenten

De bovenstaande scenario's focussen vooral op uitdagingen die inherent zijn aan LLM's zoals nauwkeurigheid en consistentie. Een probleem dat zich niet beperkt tot LLM's, maar een uitdaging kan zijn bij de verdere technologische vooruitgang in het onderwijs is het verliezen van interactie tussen studenten en docenten. Limna et al. (2023) stellen vast dat studenten en docenten persoonlijke interactie nog steeds zeer belangrijk vinden. Dit onderzoek vond dat studenten en docenten vinden dat persoonlijke interacties leiden tot een groter vertrouwen tussen beiden en deze interacties betere antwoorden op de vragen van studenten mogelijk maken. De geïnterviewde studenten vonden dat de interactie met hun docenten een persoonlijke aanpak is die niet door een chatbot vervangen kan worden. Verder stelden de studenten zelf voor dat een chatbot het best als extra hulpmiddel gebruikt wordt en interacties met docenten niet vervangt. Zij vonden dat chatbots gebruikt kunnen worden voor simpele vragen, maar dat docenten beschikbaar moeten zijn voor moeilijkere vragen en een meer persoonlijke aanpak.

Ook Mendoza et al. (2020) implementeerden een chatbot die studenten en docenten moet bijstaan. Zij ontwikkelden een webapplicatie die als tussenpersoon dient voor studenten en docenten. Ook zij besteden aandacht aan de positie die de chatbot moet innemen. Het is de bedoeling dat het een extra tool wordt, maar geen vervanging voor interactie tussen studenten en docenten. Ten slotte voerden Assayed et al. (2024) een kwalitatieve studie uit bij studenten in een middelbare school. Zij concludeerden dat korte, correcte antwoorden de voorkeur kregen op een uitgebreid gesprek waar mogelijks fouten insluipen. Zij vonden ook dat het emotionele aspect voor de studenten minder belangrijk was.

2.3 Conclusie

Uit de literatuurstudie blijkt dat AI potentieel heeft om onderwijsinstellingen te versterken door zowel studenten als docenten te ondersteunen. Niettemin worden er nog uitdagingen geïdentificeerd, met name de betrouwbaarheid van gegenereerde antwoorden en de onduidelijkheid omtrent evalueren. Een even cruciaal aandachtspunt betreft het optimaliseren van de interactie, zodat studenten de tool graag gebruiken zonder dat dit afbreuk doet aan de interacties met docenten. In hoofdstuk 3 worden enkele concepten van de literatuurstudie in de praktijk getest om de basis te leggen voor de ontwikkeling van het prototype in hoofdstuk 4.

Hoofdstuk 3

Exploratie van LLM's en ontwikkeling van het concept

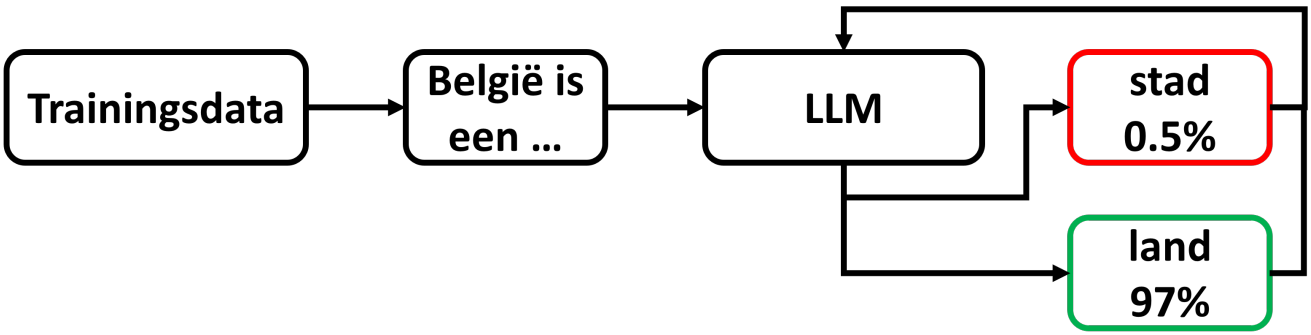
In dit hoofdstuk worden LLM's verder onderzocht. Dit begint met een theoretische achtergrond in de training en werking van LLM's. Hierna worden enkele experimenten uitgevoerd waarrond een concept ontwikkelt wordt voor het uiteindelijke prototype.

3.1 Architectuur van Large Language Models

Zoals besproken in de inleiding zijn LLM's bekend geworden door de komst van ChatGPT. Na ChatGPT is de markt gegroeid door de ontwikkeling van verschillende andere chatbots. Veel grote technologiebedrijven zoals Google (Pichai, 2023), Microsoft (Mehdi, 2023) en Meta (Meta, 2023) brachten hun eigen LLM uit. Vanwege de vele nieuwe mogelijkheden met deze technologie, ligt deze aan de basis van deze masterproef. Hierna volgt eerst een korte introductie van standaard Language Models, gevolgd door een beschrijving van de unieke eigenschappen van Large Language Models.

Een Language Model (LM) is een computationeel algoritme dat de volgorde van woorden leert op basis van statistische waarschijnlijkheden. Zo een model heeft als doel tekst te genereren, aan te vullen of te begrijpen, en dit alles met behulp van een verzameling regels en patronen die zijn afgeleid van de trainingsdata (Jurafsky and Martin, 2023).

Het onderscheidende kenmerk van Large Language Models (LLM's) zoals GPT-3 en GPT-4, ligt in hun capaciteit om een breed scala aan onderwerpen te begrijpen en te genereren. Zo werd GPT-4 getest op onder andere examens rond onderwerpen als rechten, biologie en geschiedenis (OpenAI, 2023b; Jiang et al., 2020). Dit wordt bereikt door ze te trainen met aanzienlijke hoeveelheden gevarieerde data. Het trainen van deze modellen omvat het verwerken van grote hoeveelheden tekstuele informatie, waardoor het model een diepgaand begrip ontwikkelt van taalstructuren en contexten (Tamkin et al., 2021). Een voorbeeld van deze trainingsmethodologie is beschreven voor de voorganger van de huidige modellen, GPT-2. Dit specifieke model is getraind op ongeveer 40GB aan internetteksten (Radford et al., 2019). GPT-3 daarentegen is getraind op 570 GB aan tekst, waardoor deze een veel grotere kennis heeft dan zijn voorganger (Tamkin et al., 2021).



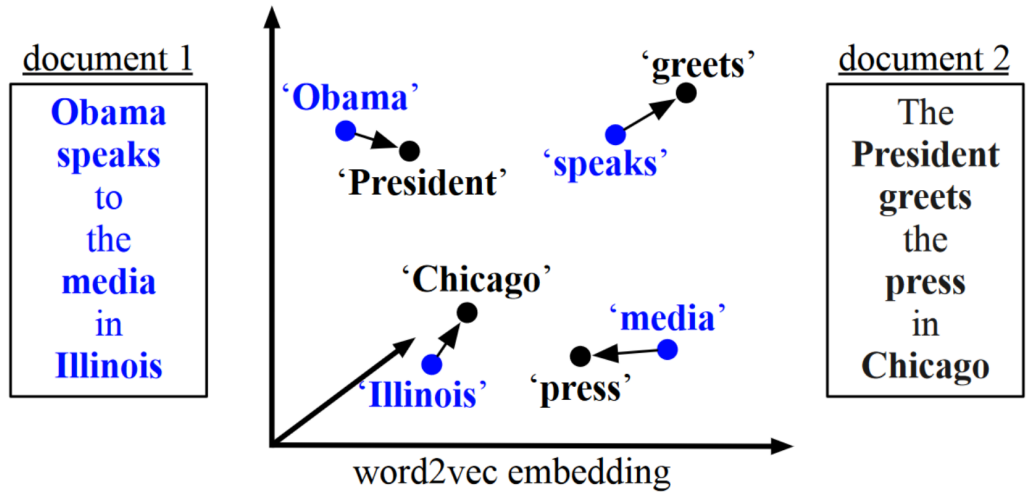
Figuur 3.1: Vereenvoudigde weergave van het trainen van een LLM. Het LLM moet het correcte vervolg op een gegeven tekst uit de trainingsdata voorspellen.

Natuurlijke taal wordt door LLM's verwerkt door teksten in tokens te verdelen, waarbij elk token meestal overeenkomt met één woord. Tijdens training voorspelt het LLM het volgende token op basis van de reeks voorgaande tokens. Dit vereist een diep begrip van taalstructuren en context, waardoor het model coherente en relevante tekst kan genereren. De uitdaging ligt in het nauwkeurig voorspellen van het logische vervolg op basis van de aangeboden tokens. Deze methode maakt LLM's waardevol voor diverse taalgerelateerde toepassingen (Radford et al., 2019). Een vereenvoudigde voorstelling van het trainen van een LLM wordt getoond in Figuur 3.1. De figuur toont hoe het begin van een zin gegeven wordt en het LLM meestal het juiste vervolg voorspelt.

De huidige LLM's zijn vooral gebaseerd op het Transformer-model (Vaswani et al., 2017). Dit model werd ontwikkeld in 2017 en wordt sindsdien veel gebruikt binnen het domein van Natural Language Processing (NLP). NLP vormt een deelgebied binnen computerwetenschappen dat vooral bezig is met computers de mogelijkheid te geven om natuurlijke taal te begrijpen en verwerken (Chowdhary, 2020). LLM's zijn binnen dit domein interessant omdat ze efficiënt grote hoeveelheden tekstdata in parallel kunnen verwerken. Verder heeft dit model ook een groot begrip van context in natuurlijke taal, wat uiterst belangrijk is binnen NLP. Het Transformer-model steunt op twee principes: *word embeddings* en *attention mechanisms* (Vaswani et al., 2017; Ramponi, 2023).

Word embeddings zijn een manier om een woord voor te stellen in een digitale omgeving (Kusner et al., 2015). De embedding wordt gebruikt in tekstanalyse om een woord voor te stellen in een vector van reële getallen. In de context van word embeddings wordt elke dimensie in de vectorruimte geassocieerd met een bepaald aspect van de betekenis van een woord. Belangrijk is dat woorden met vergelijkbare betekenissen de neiging hebben om dicht bij elkaar te liggen in deze vectorruimte (Jurafsky and Martin, 2023; Vaswani et al., 2017). Figuur 3.2 toont een voorbeeld waarbij de woorden van twee verschillende teksten in de vectorruimte worden voorgesteld. Woorden die een gelijkaardige betekenis hebben, zoals *media* en *press*, liggen dichter bij elkaar.

Attention mechanisms vormen een cruciaal onderdeel van Large Language Models (LLM's) doordat ze de mogelijkheid bieden om menselijke aandacht na te bootsen in taalverwerkingstaken. In feite stellen attention mechanisms LLM's in staat om zich te concentreren op specifieke delen van de invoer tijdens het verwerken van tekst. Het proces van attention binnen een LLM houdt in dat voor elk token in de huidige verwerkingssessie een zogenaamde *attention score* wordt berekend. Deze score geeft aan hoeveel aandacht aan elk token moet worden besteed op basis van de context



Figuur 3.2: Word emebddings van twee verschillende teksten. Woorden met vergelijkbare betekenissen liggen dichter bij elkaar (Kusner et al., 2015).

van de gehele invoer. Door deze dynamische toewijzing van aandacht kunnen LLM's prioriteit geven aan bepaalde delen van de invoer, afhankelijk van hun relevantie voor het begrijpen en genereren van de huidige output (Jurafsky and Martin, 2023; Vaswani et al., 2017; Ramponi, 2023).

3.1.1 Open-source LLM's

Het opbouwen van deze architectuur en het trainen van zulke modellen is niet omvat binnen deze masterproef vanwege de grote computationele benodigdheden. Dit wordt verder besproken in sectie 5.1. Er is echter wel een LLM nodig om de applicatie te kunnen testen en evalueren. Hiervoor wordt een beroep gedaan op open-source LLM's, wat betekent dat deze LLM's lokaal uitgevoerd kunnen worden. Hieronder worden twee voorbeelden van open-source LLM's besproken.

Het eerste voorbeeld van zo een open-source model is GPT4All (Anand et al., 2023), dat een methode biedt om gefinetunede modellen te creëren op basis van andere open-source modellen. GPT4All biedt ondersteuning voor verschillende architecturen, waaronder Falcon, LLaMa, MPT, en GPT-J. De resulterende gefinetunede modellen kunnen vervolgens lokaal worden uitgevoerd door individuen met een computer die over voldoende rekenkracht beschikt. Kleinere modellen kunnen zelfs al uitgevoerd worden op een computer met slechts 8 GB aan RAM-geheugen. Bovendien ondersteunen deze modellen ook het gebruik van een CPU voor uitvoering, waardoor een GPU niet essentieel is, hoewel deze natuurlijk wel de prestaties kan verbeteren. Deze toegankelijkheid en flexibiliteit in hardwarevereisten maken het model breed toepasbaar en minder afhankelijk van hoogwaardige hardware voor zijn functionaliteit.

Een ander voorbeeld van een open-source platform is Hugging Face (Jain, 2022), dat een grote verzameling AI-modellen omvat, waaronder LLM's, en de bijbehorende datasets voor het zelf trainen van deze modellen. Alle modellen binnen Hugging Face zijn open-source en kunnen daarom lokaal worden uitgevoerd. Bovendien biedt het platform een uniforme API-structuur voor elk model, wat de mogelijkheid biedt om per vakgebied verschillende modellen te selecteren. Een voorbeeld hiervan is een specifiek model binnen Hugging Face dat gericht is op het beschrijven van de chemische benamingen en structuren van moleculen (Christofidellis et al., 2023).

Deze diversiteit aan beschikbare modellen en de gestandaardiseerde API maken het mogelijk om specifieke expertisegebieden te benaderen met aangepaste modellen, waardoor de toepassing ervan in verschillende vakgebieden kan worden geoptimaliseerd.

3.2 Oefeningen oplossen met LLM's

Om de uitdagingen uit de literatuurstudie beter te begrijpen worden hieronder enkele experimenten uitgevoerd door LLM's vragen te stellen uit een cursus. Er wordt specifiek gekeken naar de nauwkeurigheid van de antwoorden, zoals eerder besproken in sectie 2.2.2. Er wordt een diverse set aan vragen gesteld aan ChatGPT-3.5 (ChatGPT, 2024) en Bing AI (Bing AI, 2024). Er werd gekozen om twee LLM's te gebruiken omdat de antwoorden tussen LLM's vaak grote verschillen kunnen tonen door de specifieke tuning die bedrijven kunnen toepassen. BingAI is ook gebaseerd op GPT-4, terwijl de gebruikte versie van ChatGPT een doorontwikkeling is van GPT-3. Er wordt geanalyseerd hoe een LLM elke soort vraag beantwoord en eventuele fouten in het antwoord worden toegelicht. In totaal zijn negen oefeningen door beide LLM's behandeld.

Voor de volgende secties wordt dankbaar gebruik gemaakt van de oefeningen uit de cursus Algoritmen en datastructuren (Universiteit Hasselt, 2024a). Het onderwijsteam heeft deze beschikbaar gesteld, op voorwaarde dat de volledige vragen en antwoorden niet gepubliceerd worden. Bij het evalueren van de vragen hebben zij ook bijgedragen met waardevolle feedback over de gepresenteerde antwoorden in een mondeling interview.

3.2.1 Experimenten met oefeningen

Voor de eerste oefening is gekozen voor een vraag waarbij het LLM een aanbeveling moet doen op basis van een probleem: de student moet in C++ een woordenboek creëren van een gegeven lijst woorden en hun betekenissen.

Dit probleem los je typisch op met een hashmap. ChatGPT beantwoorde de vraag ook zoals verwacht. ChatGPT toont ook een codevoorbeeld met het gebruik van een unordered_map uit de C++ Standard Library. Bing AI, in tegenstelling, koos voor een antwoord dat men eerder zou verwachten voor een oplossing in C. Er werd een voorbeeld gegeven van een implementatie van een sorted linked list. Dit geeft over het algemeen echter slechtere prestaties op operaties die voor een woordenboek vaak voorkomen, namelijk een woord toevoegen of opzoeken.

Ook bij een andere vraag is hetzelfde fenomeen te zien. Wanneer gevraagd wordt om een voorbeeld te geven van memoization bij recursieve algoritmes geven beide LLM's het klassieke voorbeeld waarbij de rij van Fibonacci berekend wordt. ChatGPT gebruikt een vector uit de C++ Standard Library om de resultaten van tussenstappen op te slaan. Bing AI gebruikt ook hier weer een gewone array van lengte 1000 om de resultaten van tussenstappen op te slaan. Dit betekent dat voor kleine rijen er veel geheugen gealloceerd wordt zonder het effectief te gebruiken en bij grote rijen de kans bestaat dat er te weinig geheugen gealloceerd is.

De geteste oefeningen tonen samen met de eerdere literatuurstudie in sectie 2.2.2 aan dat LLM's niet blindelings vertrouwd kunnen worden. Er worden te veel kleine fouten gemaakt die een student mogelijks niet zou opmerken, maar wel een belemmering zouden zijn voor het leerproces. Een mogelijke oplossing hiervoor is *prompt engineering*. Dit wordt verder besproken in sectie 3.2.2.

Figuur 3.3: Verkeerde oplossing van een vermenigvuldiging door ChatGPT-3.5. Het correcte antwoord is 38709.

De volgende oefeningen bevatten getallen en/of verwachten een getal als antwoord. Een voorbeeld van zulke vraag was eerder ook al te zien in Figuur 2.2, waar gevraagd werd om voor een gesorteerde reeks van 7 getallen de tijdscomplexiteit van lineair en binair zoeken te bepalen, zowel in het beste als slechtste scenario. ChatGPT maakt hier een fout door de tijdscomplexiteit als een letterlijke formule te zien en deze proberen uit te rekenen om het aantal stappen te vinden. Ook Bing AI had moeite met deze vraag. Hier werd als antwoord gegeven dat binair zoeken niet mogelijk was omdat de reeks niet gesorteerd is, wat wel het geval was.

Een student kan een LLM ook gebruiken om zijn oefening na te kijken. Zo is er een oefening getest waar er gevraagd wordt om voor een gegeven lijst getallen alle paren te vinden met een gegeven verschil. Wanneer aan ChatGPT en Bing AI gevraagd wordt om de oplossing voor een gegeven verschil te vinden zijn de resultaten niet bruikbaar. Afhankelijk van het gevraagde verschil tussen getallen kunnen de LLM's bijna volledig correcte en volledig verkeerde antwoorden geven. De verkeerde antwoorden waren zowel combinaties van getallen die niet het vereiste verschil gaven of getallen die niet in de gegeven rij stonden.

De oorzaak van de problemen met getallen ligt in de manier waarop LLM's getraind worden. Zoals besproken in sectie 3.1 worden LLM's getraind op het voltooien van zinnen. Hierbij wordt gefocust op een semantisch correcte zin en niet een zin waarin de feiten correct zijn. Een voorbeeld hiervan wordt getoond door Figuur 3.3. Doordat dit veroorzaakt wordt door de trainingsmethode, zal het verder niet behandeld worden in het prototype.

3.2.2 Prompt engineering

Prompt engineering is een manier om de antwoorden van LLM's in een bepaalde richting te sturen zodat deze nuttiger zijn voor de gebruiker. Dit kan al op een simpele manier door extra context te geven. Dit is ook een strategie die door OpenAI wordt aanbevolen om betere resultaten te krijgen (OpenAI, 2024). Een voorbeeld hiervan is de vraag "Hoe tel ik getallen op in Excel?". De gebruiker kan beter vragen "Hoe tel ik een rij met eurobedragen op in Excel? Ik wil dit automatisch doen voor een heel blad met rijen, waarbij alle totalen rechts eindigen in een kolom met de naam 'Totaal'.". Een ander voorbeeld is bijvoorbeeld "Schrijf een TypeScript-functie om de Fibonacci-reeks efficiënt te berekenen. Geef uitgebreid commentaar bij de code om uit te leggen wat elk onderdeel doet en waarom het zo geschreven is." teschrijven in plaats van "Schrijf code om de Fibonacci-reeks te berekenen." OpenAI (2024).

Onderzoek toont aan dat prompt engineering echter geen eenvoudig proces is (Zamfirescu-Pereira et al., 2023). Dit onderzoek vond dat gebruikers die weinig kennis hebben over programmeren en/of LLM's ook veel moeite hebben om correcte prompts te ontwikkelen. De grootste oorzaken hiervoor zijn de generalisaties die gemaakt worden door de gebruiker op basis van een enkele

observatie en het voorstellen van een gesprek met een LLM als een gesprek met een andere persoon. Er wordt dan ook onderzoek gedaan naar tools die kunnen helpen bij het ontwikkelen van prompts. Een voorbeeld is de tool ontwikkeld door Arawjo et al. (2023). Hiermee kunnen prompts op een visuele manier opgesteld en geëvalueerd worden. Het maakt het ook mogelijk om de prompts bij verschillende LLM's te testen en de verschillende antwoorden te vergelijken.

3.3 Ontwikkeling van het concept

Op basis van de literatuurstudie en de bevindingen uit sectie 3.2.1 wordt er nu een concept ontwikkelt voor het prototype met als hoofddoel te tonen hoe AI en LLM's in het speciaal kunnen bijdragen aan het onderwijs. Er wordt gestart vanuit het verantwoord integreren van een LLM. Vervolgens worden complementaire en aanvullende functies toegevoegd zodat het prototype de verschillende mogelijkheden van LLM's in het onderwijs kan tonen en een meerwaarde kan betekenen voor zowel de studenten als het onderwijsteam. Hieronder worden de verschillende doelstellingen van het prototype verder toegelicht.

De eerste doelstelling, ***D1.*** *Geef studenten op een verantwoorde manier toegang tot een LLM*, moet docenten meer controle geven over de LLM's die studenten gebruiken. Hier voor wordt gesteund op prompt engineering, zoals eerder besproken in sectie 3.2.2, om de antwoorden van de LLM's te sturen in een richting die zo goed mogelijk aansluit bij de cursus. Hier wordt specifiek gekeken naar de uitdagingen die zijn vastgesteld in hoofdstuk 2 en sectie 3.2.1.

De tweede doelstelling, ***D2.*** *Creëer een dynamische FAQ*, heeft als doel een FAQ te maken die zichzelf aanvult met vragen van studenten. Dit zal steunen op keyword extraction en sentence similarity, zoals besproken in sectie 3.3.1. Een dynamische FAQ kan een cruciale rol spelen in het ondersteunen van studenten tijdens hun leerproces. Het biedt niet alleen een gestructureerde en toegankelijke bron van informatie, maar kan ook dienen als een aanvulling op het reguliere lesmateriaal. Door gerichte vragen te beantwoorden en extra uitleg te geven, kan een FAQ studenten helpen bij het begrijpen en toepassen van de leerstof. Bovendien evolueert een dynamische FAQ mee met nieuwe vragen en uitdagingen die studenten tegenkomen, waardoor het een actueel naslagwerk wordt gedurende het hele studietraject.

De derde doelstelling, ***D3.*** *Geef aanbevelingen aan studenten*, wil op basis van verzamelde data aanbevelingen doen. Er kunnen bijvoorbeeld oefeningen aanbevolen worden om een moeilijk onderwerp te herhalen. Hiervoor worden recommender systems gebruikt, welke verder besproken worden in sectie 3.3.2. Er wordt hiervoor in het prototype gefocust op een reactief ontwerp. Een bijkomende oplossing, besproken in sectie 5.2.2, exploreert hoe de gevonden keywords gebruikt kunnen worden om oefeningen aan te bevelen die nog niet door anderen beoordeeld zijn.

De vierde en laatste doelstelling, ***D4.*** *Implementeer een transparante applicatie*, heeft als doel een applicatie te ontwikkelen voor zowel studenten als docenten die de eerste drie doelstellingen bevat. Verder moet het transparant zijn naar de gebruiker toe welke onderdelen gebaseerd zijn op AI en waarom bepaalde keuzes gemaakt zijn door deze AI-modellen.

3.3.1 Keyword extraction en sentence similarity met LLM's

Keyword extraction is een procedure binnen Natural Language Processing die gericht is op het identificeren en isoleren van de meest betekenisvolle woorden of zinsdelen in een gegeven tekst

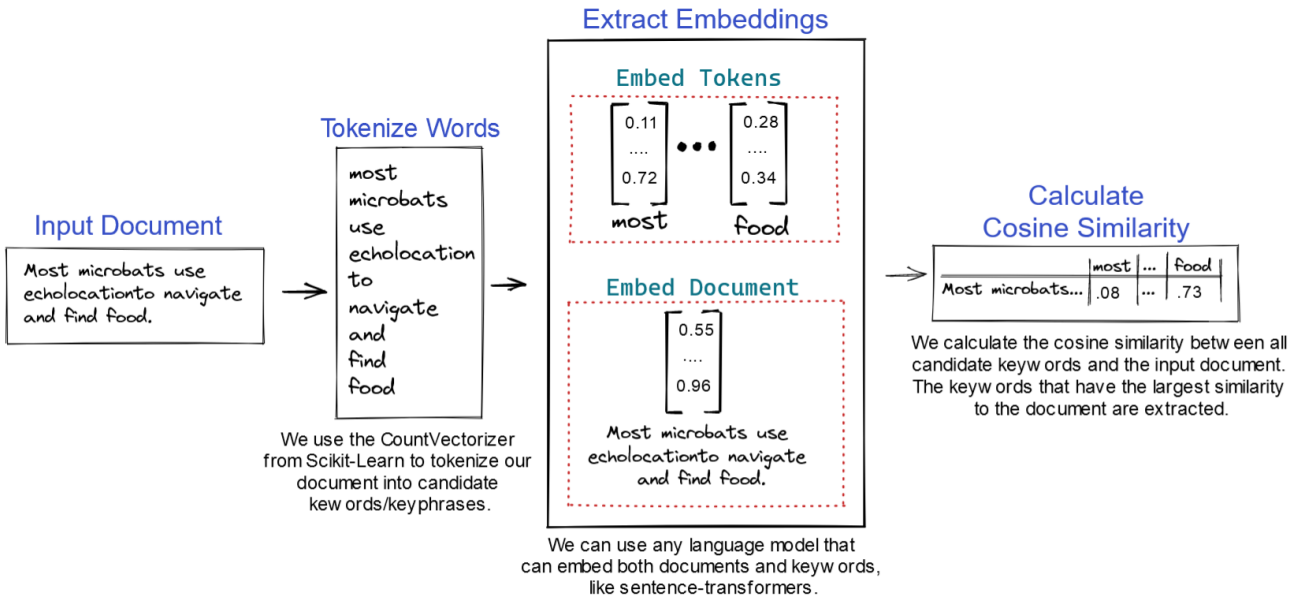
(Firoozeh et al., 2020). Deze keywords, of verzameling van keywords, worden een ngram genoemd. Een ngram is een reeks van keywords die evenveel bijdragen aan de betekenis van een tekst en die samen een goede representatie zijn van een deel van de tekst (Beliga et al., 2015).

Deze techniek heeft brede toepassingen binnen verschillende domeinen, bijvoorbeeld in zoekmachines en recommender systems. In zoekmachines worden geëxtraheerde keywords gebruikt om relevante documenten te identificeren en op te nemen in de zoekresultaten, waardoor gebruikers sneller en nauwkeuriger toegang krijgen tot informatie (Sharma and Li, 2019). Aan de andere kant worden ze in recommender systems gebruikt om de relevantie van items of content te begrijpen en passende aanbevelingen te doen aan gebruikers op basis van hun voorkeuren of zoekopdrachten (Lee and Kim, 2008).

In de academische wereld vertegenwoordigt keyword extraction een veel onderzocht onderwerp, waarbij onderzoekers streven naar verbeteringen in methoden en algoritmen om de nauwkeurigheid en efficiëntie van dit proces te verhogen. Tot de komst van LLM's waren er twee veelgebruikte technieken: statistiek en machine learning (Siddiqi and Sharan, 2015). Een voorbeeld van het gebruik van statistiek wordt gegeven door Salton et al. (1975). Hier worden woorden gerangschikt naargelang hoe goed ze verschillende documenten van elkaar kunnen onderscheiden. De woorden die bovenaan deze rangschikking eindigen worden dan verwacht goede keywords te zijn. Rose et al. (2010) heeft op basis van machine learning een algoritme ontwikkelt om keywords te detecteren. Dit algoritme steunt op een observatie dat keywords vaak meerdere woorden bevatten zonder stopwoorden of leestekens.

Recent onderzoek focust zich op de mogelijkheid om LLM's voor keyword extraction te gebruiken (Lee et al., 2023). LLM's worden typisch gebruikt om tekst te genereren vanuit een reeks tokens, maar LLM-gebaseerde keyword extraction draait dit principe om. Hierbij wordt vanuit een volledige tekst een reeks tokens gegenereerd. Deze tokens worden vervolgens voorgesteld door word embeddings. Hiermee kunnen de belangrijkste keywords bepaald worden (Sammet and Krestel, 2023). Een voorbeeld van een methode die dit soort keyword extraction gebruikt is KeyBERT. Zoals de naam al suggereert is dit model gebaseerd op *Bidirectional Encoder Representations from Transformers* (BERT), een taalmodel op basis van de eerder besproken Transformers-architectuur, ontwikkelt door Google (Devlin and Chang, 2018). BERT is open-source en kan door iedereen relatief snel getraind worden om specifieke taken uit te voeren. KeyBERT is specifiek getraind om de beste keywords uit een tekst te bepalen. Hiervoor wordt eerst een embedding gemaakt van de volledige tekst. Vervolgens wordt een embedding gemaakt van elk woord om vervolgens de embeddings met elkaar te vergelijken. Wanneer de embedding van een woord relatief dicht bij de embedding van de tekst ligt in de vectorruimte is het een goede voorstelling van de tekst en dus een goed keyword (Grootendorst, 2020). Figuur 3.4 toont het proces van KeyBERT.

Naast keyword extraction wordt ook sentence similarity gebruikt. Hiervoor wordt gekozen voor een andere tool gebaseerd op LLM's, namelijk Sentence Transformers (Reimers and Gurevych, 2019). Net zoals bij KeyBERT is de architectuur gebaseerd op het BERT LLM. Sentence Transformers maken het mogelijk om zinnen, paragrafen en afbeeldingen in de eerder genoemde vectorruimte te plaatsen. Analoog aan de keywords worden de zinnen omgezet in een embedding en embeddings die dicht bij elkaar liggen zouden een vergelijkbare betekenis moeten hebben.



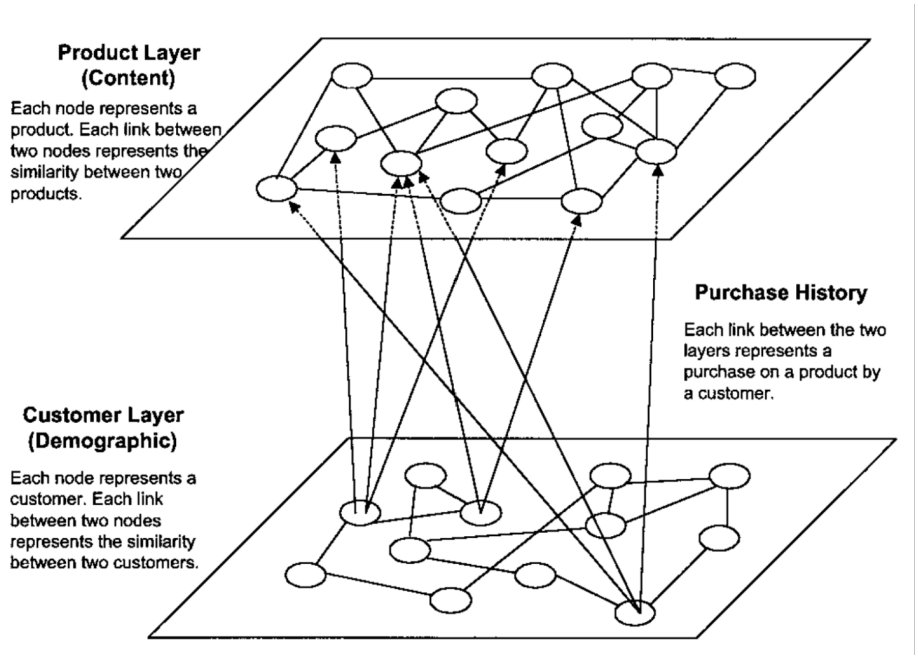
Figuur 3.4: Methode voor keyword extraction door KeyBERT. Tokens worden uit de tekst geëxtraheerd en in embeddings voorgesteld. Tokens die dicht bij de embedding van de tekst liggen zijn goede keywords (Grootendorst, 2020).

3.3.2 Recommender Systems

Een Recommender System (RS), kortweg recommender, is een systeem dat gebruikers ondersteunt bij het ontdekken van nieuwe items of diensten door middel van recommendations (aanbevelingen). Het is een systeem dat gegevens analyseert over gebruikersvoorkeuren, historisch gedrag en/of de eigenschappen van items om zo gerichte suggesties te doen (Lü et al., 2012). Of het nu gaat om het aanbevelen van boeken op basis van eerdere aankopen(Schafer et al., 2001), films op basis van kijkgeschiedenis (Gomez-Uribe and Hunt, 2016) of oefeningen op basis van eerdere moeilijkheden, het doel blijft om de gebruikerservaring te verbeteren door relevante en gepersonaliseerde aanbevelingen te bieden. Een RS werkt op basis van drie onderdelen: items, gebruikers en transacties (Lops et al., 2011).

Items vormen de kern van aanbevelingen en worden gedetailleerd beschreven aan de hand van hun unieke eigenschappen, waardoor ze onderscheidend zijn van andere items. Deze eigenschappen kunnen variëren, afhankelijk van het contextuele doel(Lops et al., 2011). Bij oefeningen kunnen eigenschappen zoals het aantal deelvragen of specifieke keywords een rol spelen bij de identificatie en selectie van de juiste items (Stanescu et al., 2013). Deze eigenschappen dienen als richtlijnen om nauwkeurige aanbevelingen te formuleren en de juiste keuzes te maken op basis van de benodigde criteria.

Gebruikers, als ontvangers van aanbevelingen, vormen een gevarieerde groep met individuele of gemeenschappelijke doelen. De manier waarop gebruikers worden vertegenwoordigd in recommender systems kan variëren en is afhankelijk van de toegepaste techniek (Lops et al., 2011). Terwijl sommige systemen eenvoudigweg gebruikmaken van volgnummers om gebruikers te identificeren (Schafer et al., 2007), gaan andere dieper door verschillende attributen te gebruiken, zoals leeftijd, beroep, voorkeuren of zelfs gedragskenmerken (Al-Shamri, 2016). Deze gedetailleerde representatie stelt recommender systems in staat om de behoeften en voorkeuren van gebruikers nauwkeuriger te begrijpen en hen relevante en gepersonaliseerde aanbevelingen te bieden.



Figuur 3.5: Voorbeeld van verbindingen tussen transacties, items en gebruikers bij de recommender voor een winkel.

Transacties zijn het logboek van interacties tussen gebruikers en de recommender, waarbij elk element wordt vastgelegd in een logboek. Deze vastgelegde transacties dienen als basis voor de verbetering van toekomstige aanbevelingen (Lops et al., 2011). Een transactie kan verschillende attributen hebben, bijvoorbeeld een specifiek item dat een gebruiker in een bepaalde context heeft gekozen. Deze informatie kan later belangrijk zijn wanneer een gebruiker met een vergelijkbaar profiel zich in een gelijkaardige context bevindt, waardoor het recommender system gerichter en preciezer kan adviseren (Wang et al., 2018). Daarnaast kan een transactie ook expliciete feedback omvatten die een gebruiker heeft gegeven over een item, zoals een beoordeling, waardoor het systeem een dieper begrip kan ontwikkelen van de voorkeuren en smaken van individuele gebruikers. De verbinding tussen items en gebruikers door transacties wordt weergegeven door Figuur 3.5.

Hoofdstuk 4

Ontwikkeling van het prototype

In dit hoofdstuk wordt het prototype uitgewerkt aan de hand van de doelstellingen uit sectie 3.3. Eerst worden de features van de ontwikkelde applicaties besproken. Hierna volgt een meer technische bespreking van elke component apart en de architectuur die deze componenten verbindt.

4.1 Ontwikkelde applicaties

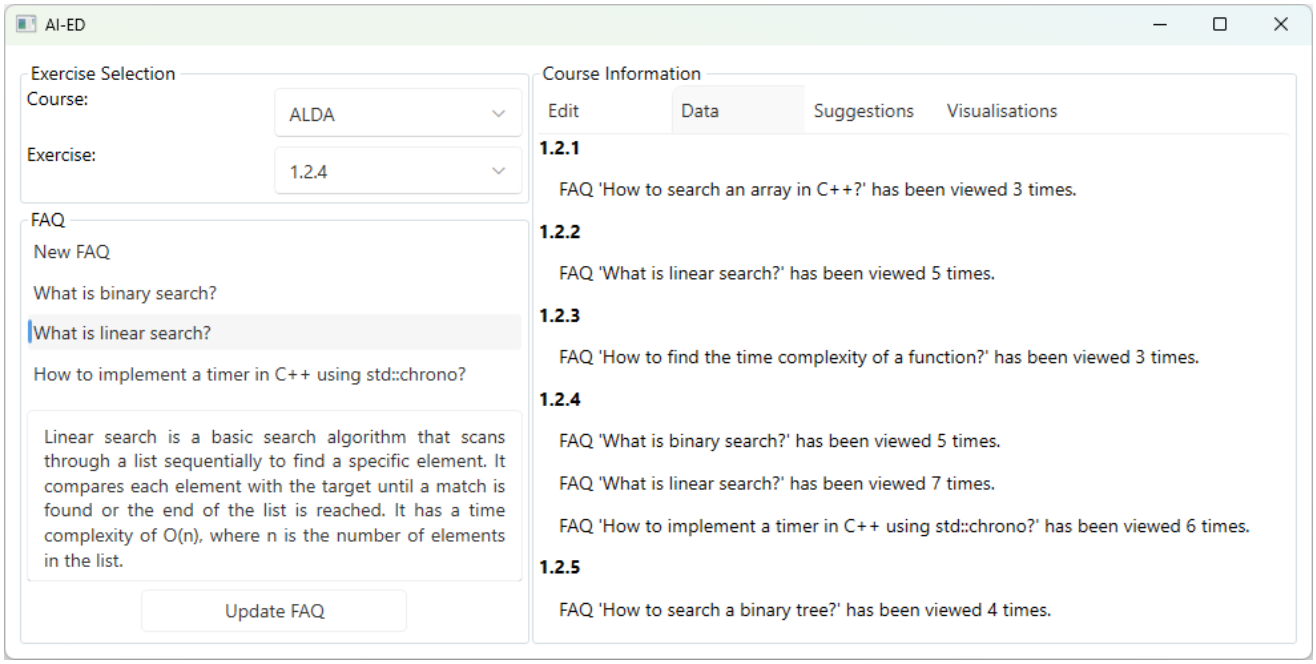
Zowel voor studenten als docenten is een specifieke applicatie ontworpen om interactie mogelijk te maken met de inhoud van de opleidingsonderdelen. De studenten kunnen bijvoorbeeld gebruikmaken van een FAQ-sectie voor het vak en vragen stellen aan een Large Language Model. Voor docenten zijn verschillende tools ontwikkeld waarmee zij de inhoud die aan studenten wordt getoond, kunnen beheren. Docenten kunnen ook data zien over de moeilijkheidsgraad van de verschillende oefeningen en de interactie van studenten en het LLM aanpassen.

4.1.1 Applicatie docenten

De docentenapplicatie biedt docenten uitgebreide controle over de cursus, het LLM en de FAQ. Verder geeft het ook inzicht in de data die verzamelt is door vragen van studenten te analyseren. Dit systeem is ontwikkeld als een Windows Presentation Foundation (WPF) applicatie, waarbij de onderliggende functionaliteit mogelijk wordt gemaakt door het gebruik van C#. Een overzicht van de applicatie wordt getoond door Figuur 4.1. Hierna wordt voor een vak stap voor stap getoond hoe een cursus wordt toegevoegd en de verschillende tools gebruikt kunnen worden.

Cursus toevoegen

Wanneer de docent voor het eerst een cursus wil toevoegen in de applicatie zijn er enkele stappen die gevolgd moeten worden. Eerst worden de gegevens van de cursus ingevuld. Buiten de naam van de cursus wordt ook de API-url ingevoerd. Deze url kan door de docent gekozen worden door verschillende modellen te testen via Hugging Face. Wanneer de url ingevoerd wordt zal de applicatie voor studenten achterliggend altijd dit model gebruiken. De docent kan dit tekstvak ook leeg laten waardoor het algemene GPT4All model gebruikt zal worden. Ten slotte kan de docent ook nog twee tekstvakken invullen met instructies voor het LLM. De eerste instructie, *Opening statement*, is een instructie die het LLM krijgt en waarvan verwacht wordt dat het LLM dit voor de rest van het gesprek onthoudt. De tweede instructie, *Every statement*, wordt met elke



Figuur 4.1: De ontwikkelde applicatie voor docenten. Links bevindt zich de dynamische FAQ en rechts suggesties met moeilijke onderwerpen.

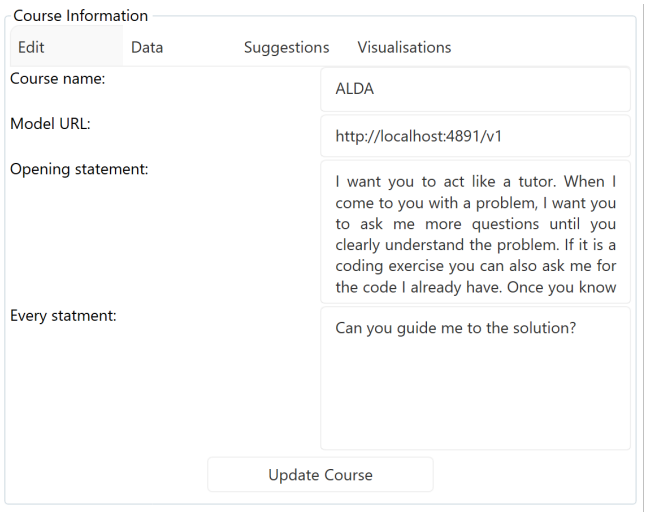
input van de student meegestuurd. Dit is, bijvoorbeeld, nuttig als de output van het LLM een bepaalde stijl moet aanhouden. Een voorbeeld van dit menu wordt getoond door Figuur 4.2a.

Keywords selecteren

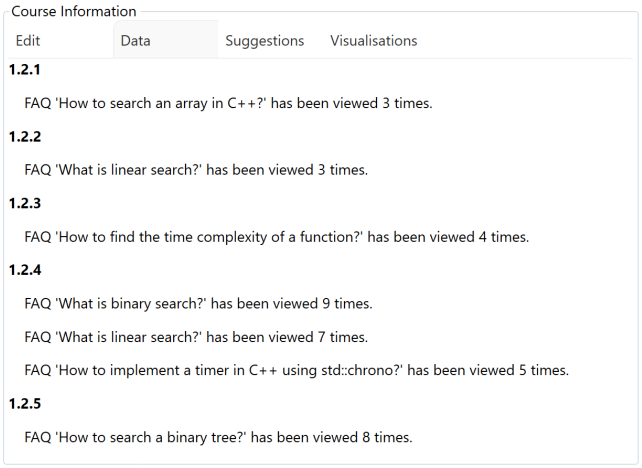
De volgende stap is het selecteren van keywords in de opgaven. Zoals uitgelegd in hoofdstuk 4.2.2 wordt dit gedaan met behulp van een LLM. Eerst wordt de docent gevraagd om de oefeningen in te voeren in een tekstvak, zoals getoond door Figuur 4.3a. De oefeningen worden gescheiden door een witregel. Voor elke oefening wordt dan KeyBERT gebruikt om de belangrijkste keywords te selecteren. Dit wordt in detail besproken in sectie 4.2.2. Omdat deze methode niet perfect is worden de gevonden keywords getoond aan de docent en kan deze zelf beslissen welke keywords behouden worden. Er kunnen indien gewenst ook keywords worden toegevoegd. Dit proces wordt getoond door Figuur 4.3b.

Verzamelde data bekijken

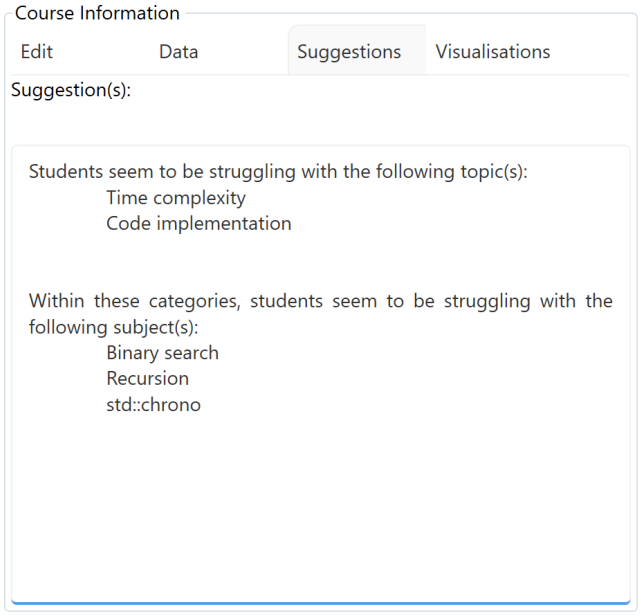
De docenten kunnen verschillende data bekijken. De eerste data zijn de gegevens van de FAQ. Zoals getoond door Figuur 4.2b, wordt hier een lijst gemaakt van alle oefeningen. Per oefening is er dan ook een onderverdeling die toont hoe vaak elke vraag in de FAQ bekeken is. De tweede data zijn de gegevens verzamelt uit de vragen van studenten, alsook de scores die ze geven aan oefeningen. Er zijn verschillende manieren om deze data te tonen. Een voorbeeld wordt getoond door Figuur 4.2d. Hierin wordt de frequentie van keywords in vragen per oefening of per vak weergegeven. Dit maakt het voor het onderwijsteam eenvoudig om moeilijke onderwerpen vast te stellen. De data met scores van oefeningen kan helpen met het opstellen of aanpassen van de cursus. Figuur 4.4a en Figuur 4.4b tonen respectievelijk het voltooiingspercentage en de ervaren moeilijkheidsgraad van oefeningen. Op basis van deze data kan het onderwijsteam bijsturen waar nodig. In het voorbeeld gegeven door deze twee grafieken kan het onderwijsteam besluiten om oefening 3 en 8 aan te passen zodat de moeilijkheid stijgt naargelang de cursus vordert.



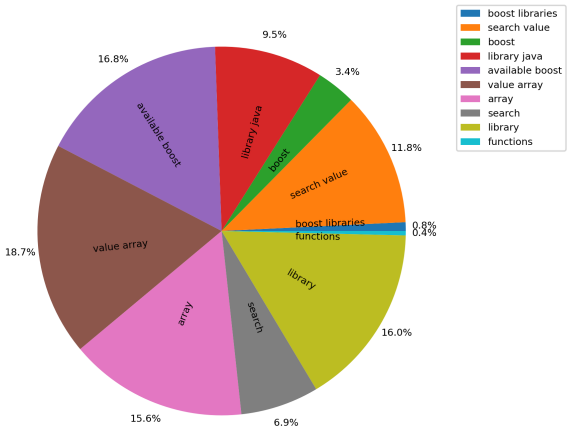
(a) Menu om wijzigen aan te brengen aan het LLM van een vak. De url kan het model zelf veranderen en de twee tekstvakken zijn instructies die het LLM ontvangt bij respectievelijk de eerste prompt en elke prompt.



(b) Verzamelde data weergegeven per oefening. Voor elke vraag in de FAQ wordt getoond hoe vaak deze bekeken is.

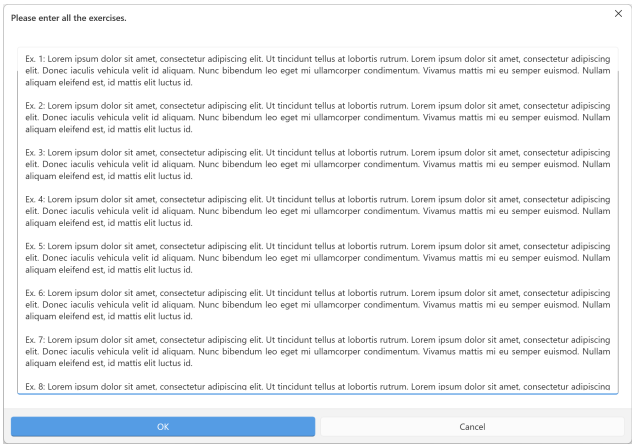


(c) Er worden suggesties getoond over moeilijke onderwerpen. De docent kan deze informatie gebruiken om de volgende les voor te bereiden.

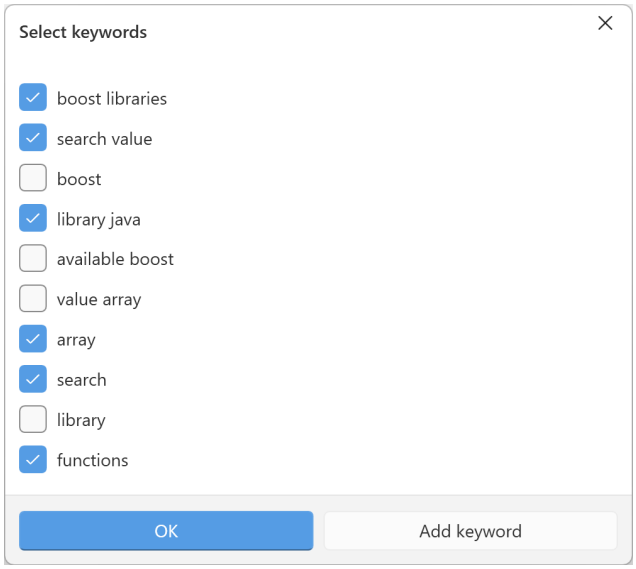


(d) Voorbeeld van een visualisatie. De docent kan diverse visualisaties zien om de verzamelde data over het vak beter te begrijpen.

Figuur 4.2: De vier menu's per vak: instellingen voor het vak (a), verzamelde data over het vak (b), suggesties voor oefeningen (c) en visualisaties van data (d).

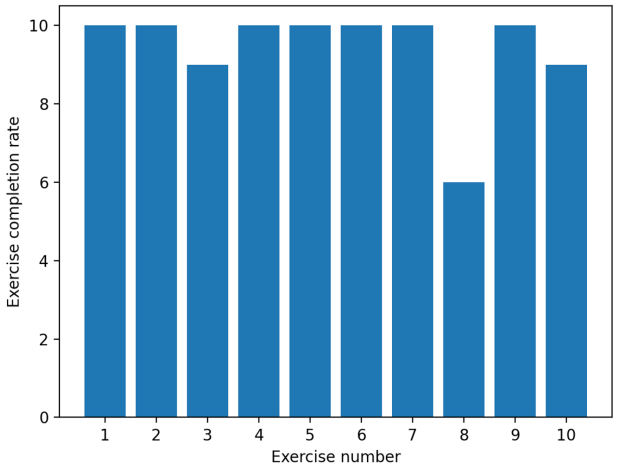


(a) De docent voert de oefeningen in zodat de keyword extraction uitgevoerd kan worden.

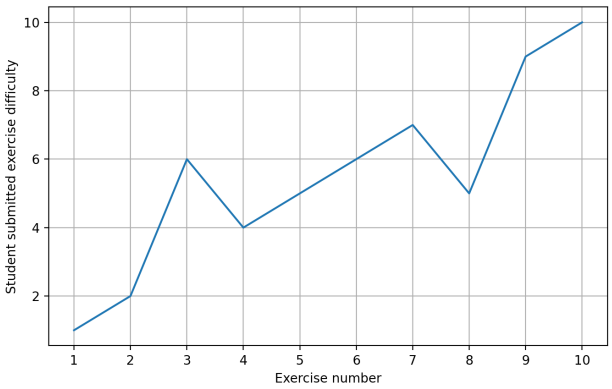


(b) De docent kan de keywords cureren of keywords toevoegen om de verzamelde data in latere stappen te verbeteren.

Figuur 4.3: Voorbeeld van de ingave van oefeningen (a) en de selectie van keywords (b).

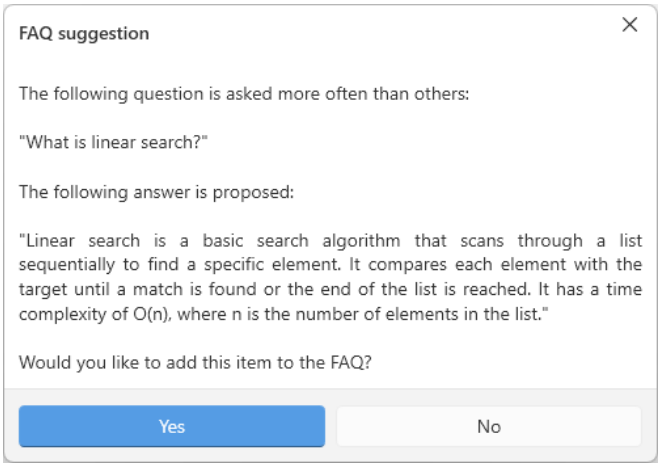


(a) Het voltooiingspercentage kan het onderwijsteam tonen welke oefeningen door weinig studenten opgelost zijn.



(b) De moeilijkheidsgraad kan het onderwijsteam meer inzicht geven in hoe de studenten de moeilijkheid van de oefeningen ervaren.

Figuur 4.4: Voorbeeld van data over het voltooiingspercentage van oefeningen (a) en de moeilijkheidsgraad van oefeningen (b).



Figuur 4.5: Voorbeeld van een suggestie om een item toe te voegen aan de FAQ. De docent kan het voorgestelde antwoord gebruiken of zelf een antwoord formuleren.

Suggesties verwerken

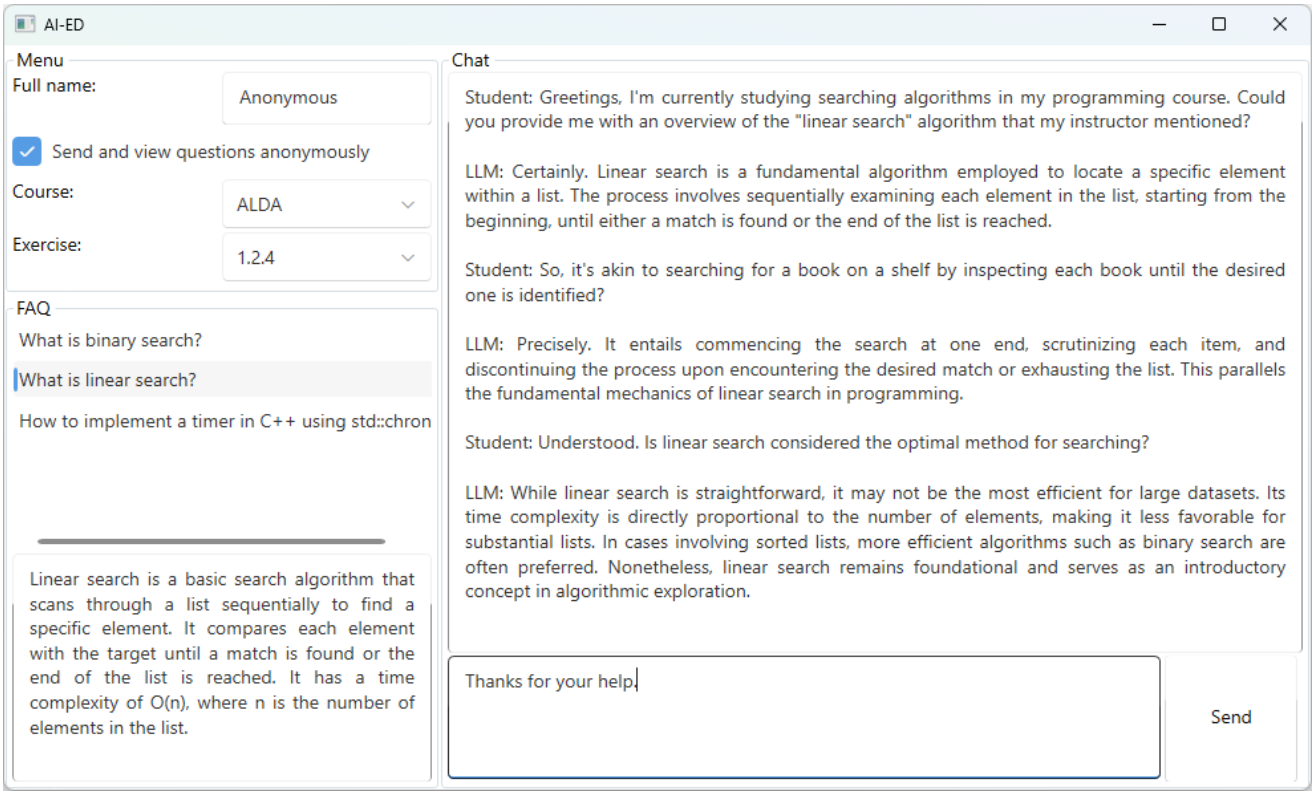
Wanneer studenten de oefeningen in de cursus maken kunnen ze ondertussen vragen stellen aan het LLM. Op deze vragen wordt ook keyword extraction uitgevoerd om het onderwerp van de vraag te begrijpen. Gelijkaardige vragen kunnen ook geïdentificeerd worden dankzij het sentence similarity model. Doordat dit model gelijkaardige vragen kan groeperen wordt het eenvoudiger om te begrijpen welke vragen veel voorkomen. Zonder dit model zou er enkel een lijst met gestelde vragen beschikbaar zijn. Veelgestelde vragen worden in een melding ook getoond aan de docent. Hierbij wordt ook een antwoord voorgesteld dat gegeven wordt door hetzelfde LLM dat de docent heeft ingesteld voor de studenten. Indien gewenst kan de docent dit antwoord ook aanpassen door de vraag manueel aan de FAQ toe te voegen. Een voorbeeld wordt getoond door Figuur 4.5.

4.1.2 Applicatie studenten

De applicatie voor studenten heeft als hoofddoel de verantwoorde toegang tot een LLM. Hiernaast biedt de applicatie ook toegang tot een dynamische FAQ en geeft de applicatie aanbevelingen voor oefeningen die de student kan maken om een bepaald onderdeel in te oefenen waarmee eerder problemen ondervonden werden. Deze applicatie, getoond door Figuur 4.6, is eveneens ontwikkeld als een Windows Presentation Foundation applicatie waar achterliggend C# gebruikt wordt. Hieronder worden alle functies individueel besproken.

Gebruik van het LLM

Studenten kunnen het LLM op een vertrouwde manier gebruiken, vergelijkbaar met hun ervaring met andere LLM-interfaces. Het proces is eenvoudig: ze kiezen een vak en hebben de mogelijkheid om al dan niet een oefening te selecteren. Door het vak te selecteren, wordt automatisch het specifieke LLM geactiveerd met de bijbehorende instructies, zoals ingesteld door de docent. Het selecteren van een oefening zorgt ervoor dat de verzamelde data gekoppeld wordt aan de juiste oefening, wat belangrijk is voor de docent om de vragen van studenten te begrijpen. Verder is dit ook belangrijk om de FAQ aan te vullen. De chat met het LLM wordt getoond door Figuur 4.6.



Figuur 4.6: De ontwikkelde applicatie voor studenten. Links bevindt zich de FAQ met rechts de chat met het LLM.

Dynamische FAQ

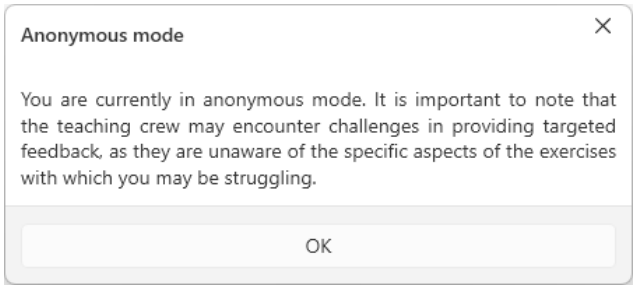
Voor de studenten kan de FAQ een hulpmiddel zijn bij het verwerken van de oefeningen. Ze kunnen voor elk vak, per oefening, verschillende vragen zien die andere studenten gesteld hebben en door de docent zijn toegevoegd. Er kunnen ook hints getoond worden die de docent kan geven om bepaalde onduidelijkheden uit de oefening weg te werken.

Anonieme data

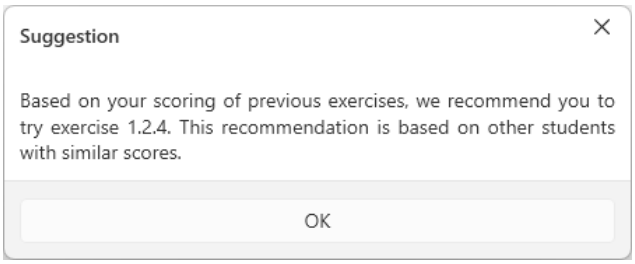
Hoewel het voor docenten handig is om per student te weten welke vragen gesteld zijn en welke oefeningen en onderwerpen deze student moeilijk vindt, is mogelijks niet elke student even comfortabel met het delen van deze data. Omdat de docenten de studenten niet willen forceren om hun naam te delen is er voor een tussenoplossing gekozen. Naast het kiezen van een vak en oefening, kan de student er ook voor kiezen om de anonieme stand aan te zetten. Zo ontvangt de docent nog altijd de data over moeilijke onderwerpen en veelgestelde vragen, maar is de naam van de student hier niet mee verbonden. De student krijgt bij het aanzetten van deze stand een melding, zoals getoond door Figuur 4.7a, die uitlegt hoe de data gebruikt wordt door de docenten om individuele feedback te geven en het aanzetten van de stand dit niet meer mogelijk maakt. Zo kan de student zelf een weloverwogen keuze maken om zijn naam te delen of niet.

Aanbevelingen van oefeningen

De laatste functie voor studenten zijn de aanbevelingen van oefeningen. Deze zijn gebaseerd op scores die de student kan geven aan oefeningen. Het profiel van de student wordt op basis van de scores gekoppeld aan gelijkaardige profielen van andere studenten. Zo kan voorspeld worden



(a) Melding om de student op de hoogte te brengen van de nadelen van het anoniem gebruik van de tool.



(b) Melding om de student een aanbeveling voor een oefening te tonen. De applicatie toont ook de bron van de aanbeveling.

Figuur 4.7: Een melding over de anonieme stand (a) en een suggestie om bijkomende oefeningen te maken (b).

welke oefeningen een voor de student moeilijk onderwerp behandelen en kunnen deze aanbevolen worden om dit onderwerp in te oefenen. Een voorbeeld van een suggestie wordt getoond door Figuur 4.7b. Het proces van deze aanbeveling wordt verder toegelicht in sectie 4.2.3.

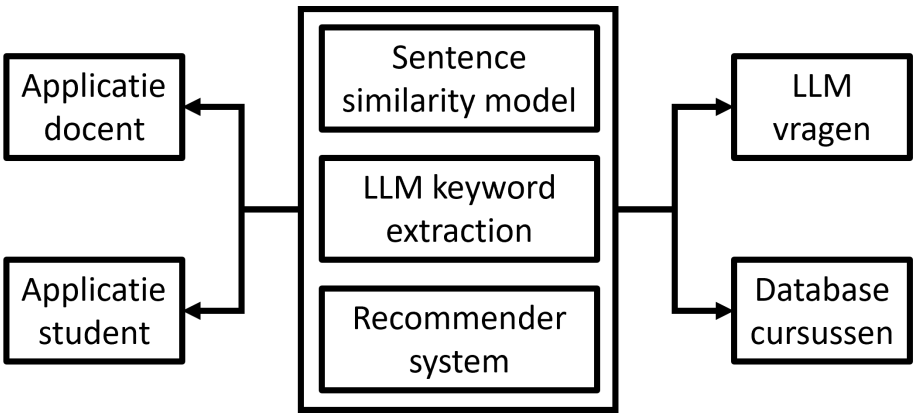
4.2 Architectuur van het systeem

De architectuur van het systeem, getoond door Figuur 4.8, is ontworpen met een sterke nadruk op het bieden van een uiterst flexibele oplossing, waardoor het prototype gemakkelijk ingezet kan worden voor diverse opleidingsonderdelen. Dit doel wordt nagestreefd door de volledige oplossing op te delen in verschillende componenten die afzonderlijk aangepast kunnen worden. Zo bestaat het prototype uit de volgende onderdelen: een LLM voor vragen van studenten (sectie 4.2.1), een bijkomend LLM voor keyword extraction en een ML-model voor sentence similarity (sectie 4.2.2), een recommender system op basis van collaborative filtering (sectie 4.2.3), logica voor de dynamische FAQ (sectie 4.2.4) en tenslotte een database voor de verschillende cursussen (sectie 4.2.5). In deze laatste sectie wordt ook de communicatie tussen de verschillende onderdelen besproken.

Om de applicaties van de studenten en docenten van gegevens te voorzien, fungeert een centrale server als knooppunt. Deze server faciliteert alle communicatie tussen de applicaties en de verschillende gebruikte tools. De diverse tools worden ingezet tijdens verschillende stadia van het toevoegen van een nieuw opleidingsonderdeel. Bij het invoeren van een cursus zal een LLM voor keyword extraction eerst de keywords per oefening bepalen. Deze keywords worden vervolgens gebruikt om de onderwerpen van vragen te identificeren en zo de FAQ dynamisch aan te vullen. Daarnaast is er een LLM beschikbaar waar studenten vragen aan kunnen stellen. Tot slot beheert de server ook de database met alle cursusinformatie.

4.2.1 Integratie LLM

De opzet van de integratie van een LLM is om de docenten meer controle te geven over welke LLM's gebruikt worden en hoe deze gebruikt worden. Ook geeft dit de mogelijkheid om data te verzamelen over veelgestelde vragen. Het streven naar extra controle wordt verwezenlijkt door gebruik te maken van een open-source model zoals besproken in sectie 3.1.1. Dit is in tegenstelling tot het inzetten van bijvoorbeeld een commercieel beschikbaar model zoals die van OpenAI.



Figuur 4.8: Architectuur van het prototype.

In dit prototype is er gekozen voor het GPT4All Falcon model. Dit model is eerder ook besproken in sectie 3.1.1 en hier is voor gekozen omdat het een brede kennis heeft over verschillende onderwerpen. Het model is echter wel geoptimaliseerd om uitgevoerd te worden op apparaten met lager computationeel vermogen (8GB RAM). Deze optimalisatie zorgt er tevens voor dat het model de snelste responssnelheid heeft van de beschikbare modellen. Een nadeel is dat het model enkel kan antwoorden in Engels.

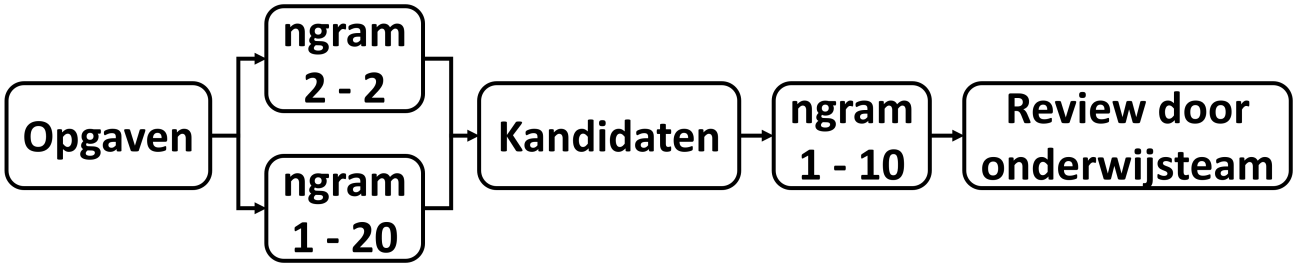
Bij de verwerking van studentenvragen worden de vragen niet direct doorgestuurd naar het door de docent gekozen AI-model. In plaats daarvan worden de vragen eerst verzameld om data te genereren die docenten kunnen analyseren en benutten om hun lessen nauwkeuriger af te stemmen op de moeilijkheden van de studenten. Deze data wordt verzameld door gebruik te maken van keyword extraction om keywords uit de gestelde vraag te bepalen, zoals uitgelegd in sectie 3.3.1. Deze keywords worden gebruikt om een overzicht te geven van veelvoorkomende onderwerpen van vragen aan het onderwijsteam. In een prototype van toekomstig onderzoek kunnen deze keywords bijvoorbeeld gebruikt worden om de recommender te ondersteunen. Dit wordt verder besproken in sectie 5.2.

De docent kan ook prompts instellen om door te geven aan het LLM. De student ziet deze niet maar ze kunnen de antwoorden wel beïnvloeden. Het model kan bijvoorbeeld bij iedere prompt geïnstrueerd worden om niet direct antwoorden te verstrekken, maar om de student te begeleiden richting de oplossing. Er kan ook een instructie ingesteld worden die slechts één keer bij de start van het gesprek gestuurd wordt. Figuur 4.2a toont hier een voorbeeld van. Het LLM wordt hier opgedragen zich te gedragen als tutor door bijvoorbeeld bijkomende vragen te stellen.

4.2.2 Keyword extraction en sentence similarity

Het vinden van keywords is de eerste stap in het toevoegen van een nieuwe cursus. Veel van de latere tools gebruiken deze keywords om verschillende taken te voltooien. Zoals besproken in hoofdstuk 3.3.1 wordt hiervoor gebruik gemaakt van KeyBERT (Grootendorst, 2020). Het volledige proces wordt getoond door Figuur 4.9.

Met behulp van KeyBERT wordt eerst een extractie gedaan van een ngram met een lengte van 1 tot 20 woorden. Concreet betekent dit dat het algoritme probeert om een embedding te maken met 1 tot 20 woorden die zo dicht mogelijk bij de embedding van de volledige oefening komt. Vervolgens wordt hetzelfde gedaan voor een ngram met specifiek twee woorden. Dit wordt



Figuur 4.9: Het proces voor keyword extraction uit de opgaven.

Oefening	**keywords**
A binary tree is a tree in which each node has zero, one, or two children. How many steps would a search for a number take in such a tree? Answer in terms of n. Note that trees will be covered in later chapters.	binary tree, number tree, tree node, tree tree, tree answer, tree, binary, zero, number, steps
In the book, they state that the time complexity of deleting from a binary search tree is typically O(log n), because deletion requires a search plus a few extra steps. Analyze the code of the delete() function to verify this.	complexity deleting, time complexity, deleting binary, logn deletion, deletion requires, complexity, deletion, tree, binary, search

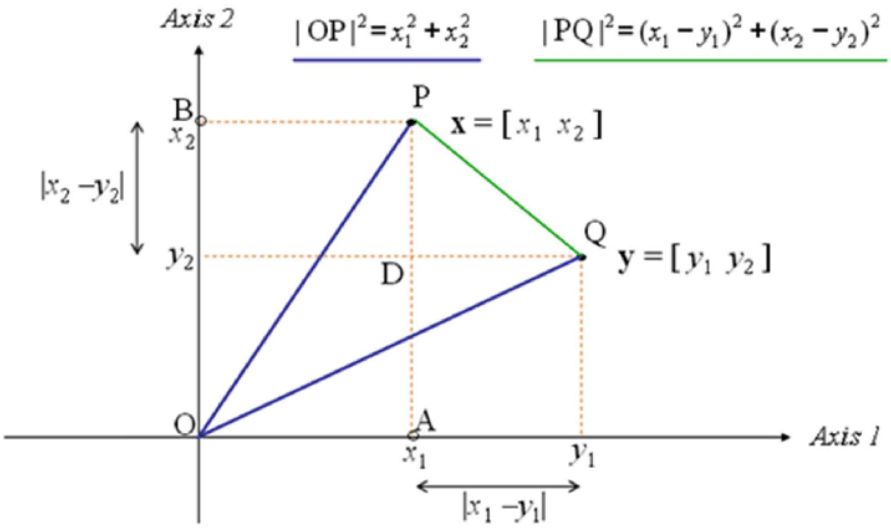
Tabel 4.1: Voorbeeld van gevonden keywords voor geselecteerde opgaven.

gedaan omdat in de oefeningen vaak twee woorden samen de volledige betekenis geven. Enkele voorbeelden zijn *time complexity*, *binary tree*, en *unordered map*.

De twee lijsten met ngrams worden vervolgens samengevoegd en gebruikt als kandidaten voor de laatste stap van het proces. Met deze kandidaten wordt het algoritme nog een laatste keer uitgevoerd en worden de uiteindelijke keywords bepaald. Voor deze laatste stap wordt een ngram ingesteld van 1 tot 10. Er wordt echter vastgesteld dat de meeste ngrammen uit maximum twee woorden bestaan. De methode om de eerder gevonden keywords als kandidaten te gebruiken in de laatste stap is experimenteel bepaald als de beste oplossing voor de geteste cursus.

Tabel 4.1 toont enkele voorbeelden van vragen met de geëxtraheerde keywords. Deze voorbeelden tonen ook een tekortkoming van het algoritme, namelijk dat sommige woorden of combinaties van woorden meer dan één keer terugkomen. Dit kan later nog wel gefilterd worden door de docent, maar het is geen probleem voor de tools die deze keywords gaan gebruiken. Het voornaamste doel is om genoeg keywords te hebben om verschillende onderwerpen van elkaar te onderscheiden. Wanneer de keywords door het algoritme zijn gevonden, worden deze getoond aan de docent en kan deze naar wens per oefening keywords verwijderen of toevoegen. Hierdoor wordt verzekerd dat de keywords die gebruikt worden in latere stappen een goede representatie zijn van de opgaven. Een voorbeeld van deze selectieprocedure door de docent wordt getoond door Figuur 4.3b.

Voor sentence similarity wordt gesteund op een gelijkaardige tool, namelijk Sentence Transformers. Hier worden verschillende zinnen omgezet naar een embedding zodat deze voorgesteld kunnen worden in een vectorruimte zoals beschreven in sectie 3.3.1. In het prototype zullen gelijkaardige vragen dicht bij elkaar liggen en kunnen ze zo gegroepeerd worden.



Figuur 4.10: Berekening van de euclidische afstand tussen twee punten in een tweedimensionale omgeving.

4.2.3 Recommender met collaborative filtering

De basisprincipes van recommenders zijn eerder al besproken in sectie 3.3.2. In deze sectie wordt dieper ingegaan op de specifieke implementatie van de recommender in het prototype. Er is gekozen voor een recommender op basis van collaborative filtering. Bij collaborative filtering worden profielen van andere gebruikers gebruikt om aanbevelingen te doen voor de huidige gebruiker. Het recommender system heeft daarvoor geen kennis nodig van de eigenschappen van de items, in dit geval de oefeningen. Sectie 5.2.2 bespreekt hoe in de toekomst eventueel keywords van de oefeningen gebruikt kunnen worden voor aanbevelingen.

Wanneer voor een student een voorspelling gedaan moet worden verzamelt het algoritme alle profielen en filtert alle studenten weg die geen enkele gemeenschappelijke oefening beoordeeld hebben. Vervolgens wordt de euclidische afstand berekent tussen de huidige gebruiker en alle andere gebruikers. Een voorbeeld voor de berekening van de euclidische afstand in een tweedimensionale omgeving wordt getoond door Figuur 4.10. De berekeningen van de recommender vinden plaatsen in een n-dimensionale omgeving, waarbij n het aantal gemeenschappelijk gescoorde oefeningen tussen beide studenten is.

Na het verzamelen van de overige profielen, wordt de verwachte score voor de oefening bepaald. Er wordt een gemiddelde score berekent, wat een benadering zou moeten zijn van de score die de student zou geven aan de oefening. Deze berekende score is een gewogen gemiddelde van de andere scores, waarbij het gewicht wordt bepaald door de euclidische afstand tussen de profielen. Studenten met profielen die dicht bij elkaar liggen, krijgen meer gewicht toegekend dan studenten waarvan de profielen ver uit elkaar liggen.

4.2.4 Dynamische FAQ

De huidige implementatie van het systeem verzamelt alle vragen die door studenten worden gesteld en ordent deze per oefening. Bovendien identificeert het systeem de belangrijkste keywords bij elke vraag. Deze keywords zijn belangrijk voor het onderwijsteam, omdat ze een dieper inzicht geven in specifieke uitdagingen die studenten ervaren bij een bepaalde opgave. Op basis hiervan

Figuur 4.11: Proces voor het aanbevelen van een item voor de FAQ.

kan het team gericht een item toevoegen aan de FAQ om deze specifieke problemen te adresseren. Dit proces, getoond door Figuur 4.11, biedt een gestructureerde aanpak om direct in te spelen op de behoeften en vragen van de studenten.

Het proces voor het identificeren van vragen vindt plaats zonder directe tussenkomst van het onderwijsteam. Dit wordt mogelijk gemaakt door het gebruik van een sentence similarity algoritme, zoals besproken in sectie 3.3.1. Met behulp van dit algoritme kunnen gelijkaardige vragen worden vastgesteld en gebundeld, waardoor een gemeenschappelijke teller kan worden bijgehouden per groep of onderwerp. Door deze gegevens kan een algoritme zelfstandig een bepaalde vraag voorstellen aan het onderwijsteam om deze toe te voegen aan de FAQ. Dit proces stelt het systeem in staat om proactief nieuwe vragen te identificeren en potentiële toevoegingen voor de FAQ voor te stellen. Belangrijk is wel dat het onderwijsteam nog steeds elk FAQ-item moet goedkeuren voor het toegevoegd wordt. In het huidige prototype wordt gebruik gemaakt van een simpele drempelwaarde voor de teller per groep. Indien deze overschreden wordt, zal het onderwijsteam de vraag als suggestie ontvangen.

4.2.5 Communicatie en database

Zoals de architectuur, getoond door Figuur 4.8, eerder al aangaf worden het sentence similarity model, het LLM voor keyword extraction en het recommender system op dit moment op één systeem uitgevoerd. Hoewel het niet noodzakelijk is om deze componenten op hetzelfde systeem uit te voeren, kan het de complexiteit wel ten goede komen. Veel van de gebruikte data voor de drie componenten is gemeenschappelijk en zou anders gedeeld moeten worden over verschillende systemen. De applicaties voor de docenten en studenten zijn wel losgekoppeld van de rest van het systeem omdat deze op het systeem van de student of docent zelf gebruikt worden. Ook het LLM met vragen en de database met cursussen is losgekoppeld van de rest van het systeem.

Het systeem dat de centrale componenten uitvoert werkt achterliggend met een Python Flask server. Hiermee is een REST (*representational state transfer*) API geïmplementeerd die de communicatie verzorgt met de andere componenten door gebruik te maken van het HTTP-protocol. REST API's hebben enkele voordelen voor het gebruik in dit prototype. Deze API's zijn *stateless* wat betekent dat de server geen informatie bijhoudt over openstaande sessies. Alle informatie moet steeds met elke nieuwe request doorgestuurd worden. Hoewel dit eerder een nadeel dan een voordeel lijkt, moet er in het achterhoofd gehouden worden dat er maar twee soorten requests zijn binnen dit prototype: opvragen van informatie en updaten van informatie. Bij het eerste type is er geen context die doorgestuurd moet worden omdat er enkel informatie wordt opgevraagd en bij het tweede type zou opgeslagen informatie ongeldig worden omdat er informatie geüpdatet wordt. Voor de uitwisseling van informatie wordt er gebruik gemaakt van het JSON-formaat.

Een bijkomend voordeel van het gebruik van een Flaskserver, is de implementatie in Python. De modellen voor sentence similarity, keyword extraction en het recommender system zijn ook allemaal in Python geïmplementeerd, waardoor er een minimale overhead is voor de communi-

catie tussen deze componenten. Hiernaast zijn er ook verschillende bibliotheken beschikbaar in Python voor het visualiseren van data. Ook hiervan is gebruik gemaakt in het prototype om de verschillende visualisaties te creëren.

Ten slotte is ook de database nog van belang. In het huidige prototype is dit geïmplementeerd als een in-memory database op hetzelfde systeem als de Flaskserver, maar dit kan in de toekomst ook verplaatst worden naar een gespecialiseerd databaseplatform. Dit zou het bijvoorbeeld mogelijk maken om de applicaties voor de docenten en studenten rechtstreeks bepaalde informatie op te vragen, terwijl deze requests nu nog door de Flaskserver moeten gaan.

Hoofdstuk 5

Reflectie

In dit hoofdstuk wordt kritisch gereflecteerd op het ontwikkelde prototype. Eerst worden de technologische beperkingen in verband met het trainen van een LLM en het implementeren van een recommender besproken. Vervolgens wordt besproken hoe het prototype verder uitgebreid of verbeterd kan worden. Ten slotte worden enkele privacyoverwegingen besproken, alsook de mogelijkheid voor een user study in toekomstig onderzoek.

5.1 Technologische beperkingen en ecologische voetafdruk

Sommige uitdagingen kunnen niet verholpen worden zonder het model opnieuw te trainen of de architectuur te wijzigen. Het trainen van een LLM is echter zo computationeel intensief dat er voor GPT-3 een geschatte 1287 MWh aan energie nodig was (de Vries, 2023). Hierbij is ook 552 ton CO_2 uitgestoten, wat overeenkomt met de uitstoot van 123 benzinewagens over de periode van een jaar (Patterson et al., 2021). Met deze beperking in het achterhoofd zijn het gebrek aan reflectie en emotie buiten beschouwing gelaten. Als we een model wel dynamisch emoties konden laten tonen had dat een interessante optie geweest voor docenten. Zoals besproken in secties 2.1.3 en 2.2.5, zijn er argumenten voor en tegen het tonen van emoties door chatbots. Docenten zouden dan zelf kunnen kiezen of emotie een bijdrage zou kunnen leveren voor hun cursus.

Het tonen van emotie is ook afhankelijk van het aantal parameters in het LLM. Wang et al. (2023) tonen aan dat vooral grote modellen een hoge score behalen voor emotionele intelligentie. Ook wordt gezien dat kleinere modellen die gefinetuned zijn een goede score kunnen behalen. Dit is te wijten aan het feit dat grotere modellen in het algemeen meer vaardigheden ontwikkelen. Het is dus moeilijk om deze vaardigheden in of uit te schakelen, zelfs als de gebruiker beschikt over de capaciteit om het model opnieuw te trainen. Het scenario, zoals eerder beschreven, waarbij docenten emotie van een LLM kunnen in- of uitschakelen is mogelijks wel te realiseren door middel van prompt engineering, maar verder onderzoek is hiervoor nodig.

Bij de recommenders is er ook het 'Cold start'-probleem aanwezig. Dit is een probleem bij recommenders waarbij er bij de start geen data is om aanbevelingen op te baseren (Bobadilla Sancho et al., 2012). Dit is een belangrijke beperking voor het prototype aangezien studenten meestal dezelfde oefeningen op hetzelfde moment gaan willen oplossen, bijvoorbeeld iedere lesweek een nieuw hoofdstuk. Om optimaal te kunnen werken zou de recommender al data moeten hebben over toekomstige oefeningen om betere aanbevelingen te kunnen genereren. Als een recommender

effectief geïmplementeerd wordt binnen een cursus, zal dit probleem na de eerste groep studenten verholpen zijn omdat er nu data is van verschillende studenten over de meeste oefeningen.

5.2 Uitbreidingen van het prototype

5.2.1 Extra ondersteuning bij LLM's en prompt engineering

Hoewel de tools voor het gebruik van LLM's al meer ondersteuning bieden, zijn er nog heel wat mogelijkheden om deze ondersteuning uit te breiden. Een voorbeeld is het implementeren van enkele standaardvragen in de chat. De docent kan dan bij bepaalde oefeningen enkele vragen instellen die aan de student gesteld worden voordat de student vragen kan stellen aan het LLM. Dit kan een alternatief zijn voor het instellen van hints in de FAQ. Door enkele weldoordachte vragen in de chat te plaatsen kan de student mogelijks al zelf een oplossing voor het probleem bedenken.

Hiernaast zou de applicatie ook meer ondersteuning kunnen bieden bij het experimenteren met prompt engineering door de docent. Dit kan bijvoorbeeld door het implementeren van een tool zoals deze ontwikkeld door Arawjo et al. (2023). Dit kan de docenten ondersteunen bij het vinden van de juiste prompts voor hun cursus en maakt de tool en prompt engineering toegankelijker voor docenten die weinig kennis hebben van deze concepten.

5.2.2 Verbeteringen aan de recommender

De recommender, zoals besproken in sectie 4.2.3, werkt op basis van collaborative filtering. Hoewel dit werkt voor het huidige prototype is er ook een meer geavanceerde oplossing mogelijk. Op basis van de gevonden keywords is het mogelijk om een recommender te implementeren die deze keywords als features gebruikt voor de oefeningen. Dit zou het mogelijk maken om oefeningen aan te bevelen aan studenten die nog niet door andere studenten beoordeeld zijn en het Cold Start-probleem, zoals besproken in sectie 5.1, gedeeltelijk kunnen oplossen doordat de recommender op basis van de keywords zelf kan besluiten of een oefening geschikt is om aan te bevelen.

Ook voor docenten kan zulk systeem een meerwaarde betekenen. Als het systeem oefeningen kan identificeren zonder dat er een score gegeven is, kan het ook pro-actief aanbevelingen doen voor, bijvoorbeeld, de volgende les. Zo zou een docent voor de les al een idee kunnen krijgen van oefeningen die studenten mogelijks moeilijk gaan vinden in de les.

5.2.3 Andere toepassingen van AI in het onderwijs

Hoewel er verschillende AI-gebaseerde tools ontwikkeld en geïmplementeerd zijn in deze masterproef, is het slechts een kleine deel uit een grote en groeiende verzameling aan AI-tools voor het onderwijs. Zo heeft Anthology, de ontwikkelaar van het Blackboard e-learningplatform, in september 2023 een update uitgebracht om docenten te ondersteunen met verschillende AI-gebaseerde tools (Anthology, 2023). Een voorbeeld hiervan is AI design assistent, die op basis van de titel en beschrijving van het vak een volledige structuur voor het vak kan opstellen, onderverdeeld in verschillende modules.

Alghamdi et al. (2020) stellen ook voor dat AI gebruikt kan worden om studenten te evalueren. Zij implementeerden een systeem dat student kan evalueren op basis van vragen die automatisch gegenereerd worden uit de cursus. Hoewel dit veelbelovend is heeft het systeem een belangrijke limitatie. Er kunnen enkel meerkeuzevragen opgesteld en geëvalueerd worden. Het probleem is dat dit soort vragen al relatief snel geëvalueerd kan worden door de docent of een assistent. Zij geven zelf ook aan dat theorievragen met een open antwoord als toekomstig werk kan dienen. Zelfs als een AI-model open vragen kan evalueren is het de vraag hoe dit ingezet kan worden. Een AI-model kan immers geen verantwoording afleggen wanneer het een fout maakt. De implementatie van zulk AI-model zou dan ook een interessant toekomstig onderzoek vormen.

Een andere toepassing is deze van een LLM als tutor. Cao (2023) stelt vast dat een LLM dat een persoonlijke leerstrategie opbouwt en gepersonaliseerde feedback kan geven op oefeningen de studenten meer het gevoel geeft dat ze thuishoren in een bepaald vak. Er wordt wel opgemerkt dat de brede kennis van LLM's toereikend genoeg kan zijn voor inleidende cursussen, maar dat er niet genoeg domeinkennis is voor meer diepgaande cursussen.

5.3 Privacy

Zoals besproken in sectie 2.2.6 is ook privacy een bezorgdheid bij het gebruik van AI en LLM's. Een voordeel van het ontwikkelde prototype is dat alles lokaal door de onderwijsinstelling beheerd kan worden. Dit betekent dat er geen data uitgewisseld wordt met een LLM-provider, zoals bijvoorbeeld OpenAI. Afhankelijk van het gebruik van Hugging Face wordt er ook data uitgewisseld, maar de modellen van Hugging Face zijn open-source en kunnen dus ook gedownload worden om lokaal uit te voeren.

Desalniettemin verzamelt de applicatie verschillende soorten data over de studenten, bijvoorbeeld gestelde vragen en gegeven scores, die allemaal opgeslagen worden. De applicatie, en in het verlengde daarvan de onderwijsinstelling, is nu verantwoordelijk voor het legaal, ethisch verwerken en opslaan van deze data. Dit moet in overeenstemming gebeuren met de nieuwe GDPR-wetgeving van de Europese Unie (Zaeem and Barber, 2020). Het moet een bewuste keuze van de studenten zijn om deze data te delen via een 'opt-in' aanpak.

5.4 User study

Hoewel het prototype het potentieel toont van een aantal AI-tools, hebben we nog geen duidelijk beeld van de eigenlijke impact van deze tools op de studenten en docenten. Om dit te onderzoeken, is er een user study nodig die de effectiviteit en gebruikservaring van de verschillende tools onderzoekt. Hieronder wordt uiteengezet hoe een dergelijke studie er zou kunnen uitzien.

Een manier om tools zoals deze te testen is door een groep deelnemers te verdelen in twee groepen: een controlegroep die op een traditionele manier studeert en een testgroep die de nieuwe tools gebruikt (Cupples et al., 1984). Om educatieve tools te testen zijn er echter nog extra overwegingen te maken bij het vinden van deelnemers. Om een conclusie te kunnen trekken over eventuele verbeteringen moet men een voldoende grote en gevarieerde groep deelnemers vinden van verschillende academische niveaus.

Vervolgens moet het studiemateriaal gekozen worden. Het materiaal moet relevant en uitdagend

genoeg zijn om de effectiviteit van de tools te testen. Enkele overwegingen bij het kiezen van het studiemateriaal zijn:

- Relevantie. Het studiemateriaal moet relevant zijn voor de deelnemers. Als het materiaal te eenvoudig of te complex is, kan dit de resultaten van de studie beïnvloeden. Het is belangrijk om materiaal te kiezen dat past bij het academische niveau van de deelnemers.
- Diversiteit. Het is ook belangrijk om een verscheidenheid aan studiemateriaal te hebben. Dit kan helpen om te zien hoe de tools presteren in verschillende contexten en onderwerpen.
- Meetbaarheid. Het studiemateriaal moet meetbare resultaten opleveren. Dit betekent dat er duidelijke criteria moeten zijn om de prestaties van de deelnemers te beoordelen.

Er moeten ook verschillende variabelen geïdentificeerd worden die gemeten worden om eventuele verschillen tussen de twee groepen vast te kunnen stellen (Vlaamse overheid, 2024). Enkele belangrijke variabelen en mogelijke onderzoeksvragen zijn:

- Transparantie. Hoe betrouwbaar zijn de voorspellingen en antwoorden en hoe goed legt de tool zijn beslissingen uit?
- Diversiteit. Past de tool zich aan de specifieke behoeften van verschillende studenten aan?
- Autonomie en controle. Beginnen studenten te veel te vertrouwen en afhankelijk te worden van de tool? Hebben de docenten nog steeds de controle en overzicht over het leerproces?
- Maatschappelijk welzijn. Legt de chatbot uit dat hij geen gevoelens of inlevingsvermogen heeft, maar deze wel nabootst? Privacy. Is het duidelijk welke gegevens verzameld worden en hoe deze verwerkt worden?

Voor de evaluatie zouden beide groepen eerst een pre-test uitvoeren om eventuele voorkennis vast te stellen. Vervolgens wordt een reeks tests afgenomen over een langere periode om begrip en retentie van de studiematerialen te meten. De resultaten van deze tests zouden dan worden vergeleken om na te gaan of er een significant verschil is in de prestaties tussen de twee groepen. Na de tests vullen de deelnemers ook een vragenlijst in, zoals bijvoorbeeld de System Usability Scale (Borsci et al., 2015; Brooke, 1996). Deze vragenlijst kan aangevuld worden om specifieke onderzoeksvragen te beantwoorden.

Het is belangrijk op te merken dat, hoewel dit onderzoeksontwerp nuttig kan zijn, er altijd rekening moet worden gehouden met mogelijke beperkingen en vooroordelen. De resultaten kunnen bijvoorbeeld worden beïnvloed door de motivatie van de deelnemers, hun eerdere ervaring met soortgelijke tools, en hun algemene studiegewoonten. Daarom is het essentieel om deze factoren in overweging te nemen bij de interpretatie van de resultaten.

Hoofdstuk 6

Besluit

Deze masterproef onderzoekt de toepasbaarheid van AI in het onderwijs, en in het bijzonder LLM's. Het onderzoek start met een uitgebreide literatuurstudie, waarin de actuele uitdagingen en toepassingen van AI en LLM's in het onderwijs worden geïdentificeerd en geanalyseerd. Op basis van deze literatuurstudie is vervolgens exploratief onderzoek uitgevoerd rond de mogelijke rol van LLM's bij het oplossen van oefeningen om vast te stellen wat moet veranderen om LLM's in te zetten in het onderwijs. De gecombineerde inzichten uit de literatuurstudie en het exploratieve onderzoek hebben geresulteerd in het formuleren van een concept voor het prototype. Dit prototype is op zijn beurt ontwikkeld om verschillende toepassingsmogelijkheden van AI en LLM's binnen het onderwijs te illustreren, met aandacht voor de perspectieven van zowel de studenten als de onderwijsteams. Door middel van het prototype worden concrete demonstraties gegeven van hoe deze technologieën kunnen worden ingezet om onderwijsuitdagingen aan te pakken.

De eerste toepassing is de integratie van een LLM zodat studenten hieraan vragen kunnen stellen. Het doel is dit op een zo verantwoord mogelijke manier te doen, rekening houdend met de eerder vastgestelde limitaties van LLM's. Dit is gerealiseerd door docenten controle te geven over welk LLM gebruikt wordt. Er is ook een optie voor de docenten om instructies aan het LLM te geven om de antwoorden in een bepaalde richting te sturen. Veelgestelde vragen worden ook voorgesteld aan de docent om toe te voegen aan een FAQ, zodat er voor studenten een actuele databank beschikbaar is met veelgestelde vragen.

Een andere toepassing is een recommender system geïmplementeerd op basis van collaborative filtering. Dit ondersteunt zowel docenten als studenten. Voor de studenten kan deze recommender oefeningen aanbevelen op basis van de oefeningen die ze eerder gemarkeerd hebben als moeilijk. Voor de docenten werkt het systeem gelijkaardig. Het kan hen helpen voorspellen welke oefeningen mogelijks een probleem vormen op basis van de eerder gemaakte oefeningen.

Ten slotte wordt er ook zo veel mogelijk data van de toepassingen beschikbaar gemaakt voor het onderwijsteam. Voor elk item in de FAQ kan de docent bijvoorbeeld bekijken hoe vaak studenten dit item bekeken hebben. Er zijn ook verschillende visualisaties beschikbaar. Zo kan de docent bijvoorbeeld inkijken welke onderwerpen het vaakst terugkomen in vragen. Deze data worden verzameld aan de hand van keyword extraction door een LLM. Verder zijn er ook grafieken beschikbaar die de moeilijkheidsgraad van oefeningen zoals ervaren door de studenten en het voltooiingspercentage van oefeningen weergeven.

De reflectie toont de verschillende technologische beperkingen van deze masterproef. Verder toont deze ook hoe het prototype in toekomstig onderzoek uitgebreid kan worden met verschillende extra functies. Er worden ook enkele andere implementaties besproken die tonen dat er nog verschillende andere mogelijkheden zijn met AI in het onderwijs. Er wordt ook besproken wat de impact is van privacy op het prototype. Ten slotte wordt er een concept opgesteld van een user study die de werking van het prototype kan evalueren.

Uit dit onderzoek kan geconcludeerd worden dat de aanpasbare LLM's, voorgestelde oefeningen en dynamische FAQ bijdragen aan de mogelijkheden om het onderwijs te verbeteren met AI. De extra data zorgt er ook voor dat de interactie tussen docenten en studenten niet verminderd en docenten hun materiaal beter kunnen afstemmen op de studenten. Deze masterproef legt ook de basis voor verder onderzoek dat zich kan richten op het uitvoeren van een user study om de effectiviteit en gebruikservaring te evalueren.

Literatuurlijst

Al-Shamri, M. Y. H. (2016). User profiling approaches for demographic recommender systems. *Knowledge-Based Systems*, 100:175–187.

Alghamdi, A. A., Alanezi, M. A., and Khan, F. (2020). Design and implementation of a computer aided intelligent examination system. *International Journal of Emerging Technologies in Learning (iJET)*, 15(01):pp. 30–44.

Anand, Y., Nussbaum, Z., Duderstadt, B., Schmidt, B., and Mulyar, A. (2023). Gpt4all: Training an assistant-style chatbot with large scale data distillation from gpt-3.5-turbo. `https://github.com/nomic-ai/gpt4all`.

Anthology (2023). What's New in Blackboard Learn Ultra – September 2023. `https://www.anthology.com/blog/whats-new-in-blackboard-learn-ultra-september-2023`. [Online; accessed 19. Jan. 2024].

Arawjo, I., Swoopes, C., Vaithilingam, P., Wattenberg, M., and Glassman, E. (2023). Chainforge: A visual toolkit for prompt engineering and llm hypothesis testing.

Assayed, S. K., Woods, D., Alkahtib, M., and Shaalan, K. (2024). Effective human-chatbot interaction: A high school student perspective. *International Journal of Computing and Digital Systems*, 15:1–9.

Beliga, S., Meštrović, A., and Martinčić-Ipšić, S. (2015). An overview of graph-based keyword extraction methods and approaches. *Journal of information and organizational sciences*, 39(1):1–20.

Bing AI (2024). Copilot met GPT-4. `https://www.bing.com/chat?q=Bing+AI&FORM=hpcodx`. [Online; accessed 13. Jan. 2024].

Bobadilla Sancho, J., Ortega Requena, F., Hernando Esteban, A., and Bernal Bermúdez, J. (2012). A collaborative filtering approach to mitigate the new user cold start problem. *Knowledge-Based Systems*, 26:225–238.

Borsci, S., Federici, S., Bacci, S., Gnaldi, M., and Bartolucci, F. (2015). Assessing user satisfaction in the era of user experience: Comparison of the sus, umux, and umux-lite as a function of product experience. *International Journal of Human–Computer Interaction*, 31(8):484–495.

Brooke, J. (1996). Sus -a quick and dirty usability scale usability and context. *Usability evaluation in industry*, 189:4–7.

Cao, C. (2023). Leveraging large language model and story-based gamification in intelligent tutoring system to scaffold introductory programming courses: A design-based research study.

Ceuppens, L. (2023). Universiteit Antwerpen onderzoekt of student paper schreef met artificiële intelligentie van ChatGPT. *vrtnws.be*.

ChatGPT (2024). ChatGPT. `https://chat.openai.com`. [Online; accessed 13. Jan. 2024].

Chen, C.-M. (2011). Web-based remote human pulse monitoring system with intelligent data analysis for home health care. *Expert Systems with Applications*, 38(3):2011–2019.

Chiu, T. K., Xia, Q., Zhou, X., Chai, C. S., and Cheng, M. (2023). Systematic literature review on opportunities, challenges, and future research recommendations of artificial intelligence in education.

Chowdhary, K. R. (2020). *Natural Language Processing*, pages 603–649. Springer, New Delhi.

Christofidellis, D., Giannone, G., Born, J., Winther, O., Laino, T., and Manica, M. (2023). Unifying molecular and textual representations via multi-task language modelling. *arXiv preprint arXiv:2301.12586*.

Cupples, L. A., Heeren, T., Schatzkin, A., and Colton, T. (1984). Multiple testing of hypotheses in comparing two groups. *Annals of Internal Medicine*, 100(1):122–129.

De Morgen (2023). Lerarentekort wordt steeds groter probleem: nu al 3.195 openstaande vacatures. *De Morgen*.

de Vries, A. (2023). The growing energy footprint of artificial intelligence. *Joule*, 7:2191–2194.

Devlin, J. and Chang, M.-W. (2018). Open Sourcing BERT: State-of-the-Art Pre-training for Natural Language Processing. `https://blog.research.google/2018/11/open-sourcing-bert-state-of-art-pre.html`. [Online; accessed 6. Jan. 2024].

EDM (2024a). Expertisecentrum voor Digitale Media - EDM - UHasselt. `https://www.uhasselt.be/nl/instituten/expertisecentrum-voor-digitale-media`. [Online; accessed 9. Jan. 2024].

EDM (2024b). Intelligible Interactive Systems - EDM - UHasselt. `https://www.uhasselt.be/nl/instituten/expertisecentrum-voor-digitale-media/onderzoek/intelligible-interactive-systems`. [Online; accessed 9. Jan. 2024].

European Commision (2024). Intellectual Property in ChatGPT. [Online; accessed 15. Jan. 2024].

Felder, R. and Silverman, L. (2002). Learning and teaching styles in engineering education. *Engineering Education*.

Firoozeh, N., Nazarenko, A., Alizon, F., and Daille, B. (2020). Keyword extraction: Issues and methods. *Natural Language Engineering*, 26(3):259–291.

Flanders AI (2024). Vlaams AI-Onderzoeksprogramma. [Online; accessed 10. Jan. 2024].

Fryer, L. K., Ainley, M., Thompson, A., Gibson, A., and Sherlock, Z. (2017). Stimulating and sustaining interest in a language course: An experimental comparison of chatbot and human task partners. *Computers in Human Behavior*, 75:461–468.

Gao, C. A., Howard, F. M., Markov, N. S., Dyer, E. C., Ramesh, S., Luo, Y., and Pearson, A. T. (2022). Comparing scientific abstracts generated by chatgpt to original abstracts using

an artificial intelligence output detector, plagiarism detector, and blinded human reviewers. *bioRxiv*.

Gilliland, N. (2016). Domino's introduces 'Dom the Pizza Bot' for Facebook Messenger. [Online; accessed 10. Jan. 2024].

Glikson, E. and Woolley, A. W. (2020). Human trust in artificial intelligence: Review of empirical research. *https://doi.org/10.5465/annals.2018.0057*, 14:627–660.

Gomez-Uribe, C. A. and Hunt, N. (2016). The netflix recommender system: Algorithms, business value, and innovation. *ACM Trans. Manage. Inf. Syst.*, 6(4).

Grootendorst, M. (2020). Keyword Extraction with BERT. *Medium*.

Grynbaum, M. M. and Mac, R. (2023). New York Times Sues OpenAI and Microsoft Over Use of Copyrighted Work. *N.Y. Times*.

Hwang, G. J., Sung, H. Y., Chang, S. C., and Huang, X. C. (2020). A fuzzy expert system-based adaptive learning approach to improving students' learning performances by considering affective and cognitive factors. *Computers and Education: Artificial Intelligence*, 1:100003.

Ijaz, K., Bogdanovych, A., and Trescak, T. (2017). Virtual worlds vs books and videos in history education. *Interactive Learning Environments*, 25.

Jain, S. M. (2022). *Hugging Face*, pages 51–67. Apress, Berkeley, CA.

Jawahar, G., Abdul-Mageed, M., and Lakshmanan, L. V. S. (2020). Automatic detection of machine generated text: A critical survey.

Jiang, Z., Xu, F. F., Araki, J., and Neubig, G. (2020). How Can We Know What Language Models Know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Johannsen, F., Leist, S., Konadl, D., and Basche, M. (2018). Comparison of commercial chatbot solutions for supporting customer interaction.

Jurafsky, D. and Martin, J. H. (2023). Speech and language processing.

Kasneci, E., Sessler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., Gasser, U., Groh, G., Günnemann, S., Hüllermeier, E., Krusche, S., Kutyniok, G., Michaeli, T., Nerdel, C., Pfeffer, J., Poquet, O., Sailer, M., Schmidt, A., Seidel, T., Stadler, M., Weller, J., Kuhn, J., and Kasneci, G. (2023). Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and Individual Differences*, 103.

Katholieke Universiteit Leuven (2024a). Verantwoord gebruik van generatieve AI in ons onderwijs. [Online; accessed 11. Jan. 2024].

Katholieke Universiteit Leuven (2024b). Verantwoord gebruik van generatieve artificiële intelligentie. `https://www.kuleuven.be/genai`. [Online; accessed 9. Jan. 2024].

Kushik, N., Yevtushenko, N., and Evtushenko, T. (2020). Novel machine learning technique for predicting teaching strategy effectiveness. *International Journal of Information Management*, 53:101488.

Kusner, M., Sun, Y., Kolkin, N., and Weinberger, K. (2015). From word embeddings to document distances. In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference*

on Machine Learning, volume 37 of *Proceedings of Machine Learning Research*, pages 957–966, Lille, France. PMLR.

Köse, U. (2017). An augmented-reality-based intelligent mobile application for open computer education. *Mobile Technologies and Augmented Reality in Open Education*, pages 154–174.

Lee, S. and Kim, H.-J. (2008). News keyword extraction for topic tracking. In *2008 Fourth International Conference on Networked Computing and Advanced Information Management*, volume 2, pages 554–559.

Lee, W., Chun, M., Jeong, H., and Jung, H. (2023). Toward keyword generation through large language models. In *Companion Proceedings of the 28th International Conference on Intelligent User Interfaces*, IUI '23 Companion, page 37–40, New York, NY, USA. Association for Computing Machinery.

Limna, P., Kraiwanit, T., Jangjarat, K., Klayklung, P., and Chocksathaporn, P. (2023). The use of chatgpt in the digital era: Perspectives on chatbot implementation. *Journal of Applied Learning and Teaching*, 6.

Lops, P., de Gemmis, M., and Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. *Recommender Systems Handbook*, pages 73–105.

Lü, L., Medo, M., Yeung, C. H., Zhang, Y.-C., Zhang, Z.-K., and Zhou, T. (2012). Recommender systems. *Physics Reports*, 519(1):1–49. Recommender Systems.

Mehdi, Y. (2023). Reinventing search with a new AI-powered Microsoft Bing and Edge, your copilot for the web - The Official Microsoft Blog. `https://blogs.microsoft.com/blog/2023/02/07/reinventing-search-with-a-new-ai-powered-microsoft-bing-and-edge-your-copilot-for-the-web`.

Mendoza, S., Hernández-León, M., Sánchez-Adame, L. M., Rodríguez, J., Decouchant, D., and Meneses-Viveros, A. (2020). Supporting student-teacher interaction through a chatbot. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12206 LNCS:93–107.

Meta (2023). Introducing LLaMA: A foundational, 65-billion-parameter language model. `https://ai.meta.com/blog/large-language-model-llama-meta-ai`. [Online; accessed 10. Jan. 2024].

Metz, C. (2024). OpenAI Says New York Times Lawsuit Against It Is 'Without Merit'. *N.Y. Times*.

Mitchell, A. (2023). AI thinks the Constitution was written by bots — but there's a bigger issue. *New York Post*.

OpenAI (2022). Introducing chatgpt. `https://openai.com/blog/chatgpt`. [Online; accessed 9. Jan. 2024].

OpenAI (2023a). Data Controls FAQ | OpenAI Help Center. `https://help.openai.com/en/articles/7730893-data-controls-faq`. [Online; accessed 1. Jan. 2024].

OpenAI (2023b). Gpt-4 technical report. `https://arxiv.org/abs/2303.08774v4`. [Online; accessed 1. Jan. 2024].

OpenAI (2024). Prompt Engineering. https://platform.openai.com/docs/guides/prompt-engineering/strategy-write-clear-instructions. [Online; accessed 18. Jan. 2024].

Ozden, A., Faghri, A., and Li, M. (2016). Using knowledge-automation expert systems to enhance the use and understanding of traffic monitoring data in state dots. *Procedia Engineering*, 145:980–986. ICSDEC 2016 – Integrating Data Science, Construction and Sustainability.

Passey, D., Shonfeld, M., Appleby, L., Judge, M., Saito, T., and Smits, A. (2018). Digital agency: Empowering equity in and through education. *Technology, Knowledge and Learning*, 23(3):425–439.

Patterson, D., Gonzalez, J., Le, Q., Liang, C., Munguia, L.-M., Rothchild, D., So, D., Texier, M., and Dean, J. (2021). Carbon emissions and large neural network training.

Peterson, C. (2003). Bringing addie to life: Instructional design at its best. *Journal of Educational Multimedia and Hypermedia*, 12:227–241.

Pichai, S. (2023). An important next step on our AI journey. https://blog.google/technology/ai/bard-google-ai-search-updates.

Radford, A., Wu, J., Amodei, D., Amodei, D., Clark, J., Brundage, M., Sutskever, I., Askell, A., Lansky, D., Hernandez, D., and Luan, D. (2019). Better language models and their implications.

Ramponi, M. (2023). The Full Story of Large Language Models and RLHF. https://www.assemblyai.com/blog/the-full-story-of-large-language-models-and-rlhf.

Reimers, N. and Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Rose, S., Engel, D., Cramer, N., and Cowley, W. (2010). Automatic keyword extraction from individual documents. *Text Mining: Applications and Theory*, pages 1–20.

Rowe, K. (2006). Effective teaching practices for students with and without learning difficulties: Issues and implications surrounding key findings and recommendations from the national inquiry into the teaching of literacy. *Australian journal of learning disabilities*, 11(3):99–115.

Saibene, A., Assale, M., and Giltri, M. (2021). Expert systems: Definitions, advantages and issues in medical field applications. *Expert Systems with Applications*, 177:114900.

Salton, G., Yang, C. S., and Yu, C. T. (1975). A theory of term importance in automatic text analysis. *Journal of the American Society for Information Science*, 26(1):33–44.

Sammet, J. and Krestel, R. (2023). Domain-specific keyword extraction using bert. In *Proceedings of the 4th Conference on Language, Data and Knowledge*, pages 659–665.

Schafer, J. B., Frankowski, D., Herlocker, J., and Sen, S. (2007). *Collaborative Filtering Recommender Systems*, pages 291–324. Springer Berlin Heidelberg, Berlin, Heidelberg.

Schafer, J. B., Konstan, J. A., and Riedl, J. (2001). E-commerce recommendation applications. *Data Mining and Knowledge Discovery*, 5(1):115–153.

Sharma, P. and Li, Y. (2019). Self-supervised contextual keyword and keyphrase retrieval with self-labelling. *Preprints*.

Siddiqi, S. and Sharan, A. (2015). Keyword and keyphrase extraction techniques: a literature review. *International Journal of Computer Applications*, 109(2).

Stanescu, A., Nagar, S., and Caragea, D. (2013). A hybrid recommender system: User profiling from keywords and ratings. In *2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, volume 1, pages 73–80.

Tamkin, A., Brundage, M., Clark, J., and Ganguli, D. (2021). Understanding the capabilities, limitations, and societal impact of large language models.

Tlili, A., Shehata, B., Adarkwah, M. A., Bozkurt, A., Hickey, D. T., Huang, R., and Agyemang, B. (2023). What if the devil is my guardian angel: Chatgpt as a case study of using chatbots in education. *Smart Learning Environments*, 10.

Tärning, B., Silvervarg, A., Gulz, A., and Haake, M. (2019). Instructing a teachable agent with low or high self-efficacy – does similarity attract? *International Journal of Artificial Intelligence in Education*, 29:89–121.

Universitair Ziekenhuis Leuven (2024). Onderzoeken nucleaire geneeskunde. `https://www.uzleuven.be/nl/nucleaire-geneeskunde/onderzoeken-nucleaire-geneeskunde`. [Online; accessed 11. Jan. 2024].

Universiteit Hasselt (2024a). Informatie opleidingsonderdeel. `https://studiegidswww.uhasselt.be/opleidingsonderdeel.aspx?a=2023&i=4900&n=4&t=01`. [Online; accessed 12. Jan. 2024].

Universiteit Hasselt (2024b). Training in radiation protection- UHasselt. `https://www.uhasselt.be/en/onderzoeksgroepen-en/nutec/application/training-in-radiation-protection`. [Online; accessed 11. Jan. 2024].

U.S. Copyright Office (2024). Chapter 1 - Circular 92 | U.S. Copyright Office. `https://www.copyright.gov/title17/92chap1.html#107`. [Online; accessed 16. Jan. 2024].

Valuates (2023a). Global and India Artificial Intelligence Software Market, Report Size, Worth, Revenue, Growth, Industry Value, Share 2023. `https://reports.valuates.com/market-reports/QYRE-Auto-30Z15530/global-and-india-artificial-intelligence-software`. [Online; accessed 9. Jan. 2024].

Valuates (2023b). Large Language Model (LLM) Market Report Size Worth Revenue Growth Industry Value Share 2023. `https://reports.valuates.com/market-reports/QYRE-Auto-30B13652/global-large-language-model-llm`. [Online; accessed 9. Jan. 2024].

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Łukasz Kaiser, and Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 2017-December:5999–6009.

Vlaamse overheid (2024). Concrete vragen om AI-programma's te evalueren, kiezen en gebruiken. `https://www.vlaanderen.be/kenniscentrum-digisprong/themas/innovatie/artificiele-intelligentie/concrete-vragen-om-ai-programmas-te-evalueren-kiezen-en-gebruiken`. [Online; accessed 19. Jan. 2024].

Wadden, J. J. (2021). Defining the undefinable: the black box problem in healthcare artificial intelligence. *Journal of Medical Ethics*.

Wagner, M. W. and Ertl-Wagner, B. B. (2023). Accuracy of information and references using chatgpt-3 for retrieval of clinical radiological information. *Canadian Association of Radiologists Journal*.

Walkington, C. and Bernacki, M. L. (2019). Personalizing algebra to students' individual interests in an intelligent tutoring system: Moderators of impact. *International Journal of Artificial Intelligence in Education*, 29:58–88.

Wang, S., Hu, L., Cao, L., Huang, X., Lian, D., and Liu, W. (2018). Attention-based transactional context embedding for next-item recommendation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Wang, X., Li, X., Yin, Z., Wu, Y., and Liu, J. (2023). Emotional intelligence of large language models. *Journal of Pacific Rim Psychology*, 17.

Wei, Y., Yang, Q., Chen, J., and Hu, J. (2018). The exploration of a machine learning approach for the assessment of learning styles changes, 121-126. *Mechatronic Systems and Control 2018*, 46:121–126.

Weizenbaum, J. (1966). Eliza - a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9:36–45.

Wu, Y. and Hu, G. (2023). Exploring prompt engineering with GPT language models for document-level machine translation: Insights and findings. In *Proceedings of the Eighth Conference on Machine Translation*, pages 166–169, Singapore. Association for Computational Linguistics.

Zaeem, R. N. and Barber, K. S. (2020). The effect of the gdpr on privacy policies: Recent progress and future promise. *ACM Trans. Manage. Inf. Syst.*, 12(1).

Zamfirescu-Pereira, J., Wong, R. Y., Hartmann, B., and Yang, Q. (2023). Why johnny can't prompt: How non-ai experts try (and fail) to design llm prompts. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, New York, NY, USA. Association for Computing Machinery.

Zhang, K. and Aslan, A. B. (2021). Ai technologies for education: Recent research & future directions. *Computers and Education: Artificial Intelligence*, 2.