

# Itineris - Ontwikkeling van een AI assistent voor Itineris: technologieën en valkuilen, Arthur Uydens (2024).

---

**Arthur Uydens.**

Scriptie voorgedragen tot het bekomen van de graad van  
Professionele bachelor in de toegepaste informatica

**Promotor:** Dhr. J. Claes

**Co-promotor:** Dhr. P. Baeke

**Academiejaar:** 2023–2024

**Eerste examenperiode**

**Departement IT en Digitale Innovatie .**



# Woord vooraf

Met deze bachelorproef wil ik een goede manier vinden om een chatbot zinvolle antwoorden te laten geven op vragen die over bedrijfsdata gaat met het gebruik van artificiële intelligentie. Ik heb voor deze bachelorproef nog maar weinig onderzoek naar artificiële intelligentie gedaan maar toch is het altijd een onderwerp geweest dat me fascineerde. Ik had verschillende bedrijven gestuurd met de vraag of ze een casus hadden waar ik op kon inwerken maar toen ik antwoord kreeg met een onderwerp van Itineris wist ik al relatief snel dat dit was waar ik onderzoek naar wou doen. Daarom wil ik dus zeker mijn co-promotor, Peter Baeke, bedanken voor mij deze kans te geven en wekelijkse ondersteuning te geven waar het kon. Mijn promotor, Jan Claes, wil ik ook hartelijk bedanken voor altijd snelle feedback te geven en te helpen met vragen die ik had over het verloop van de bachelorproef. Voor mijn meer technische vragen kon ik ook altijd terecht bij Maarten Glas en Tom Uvin die me ook doorheen het verloop feedback hebben gegeven op mijn werk. Ik wens u veel leesplezier en hoop dat u iets nuttig kan halen uit deze bachelorproef!

# Samenvatting

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

# Inhoudsopgave

<b>Lijst van figuren</b>	<b>vii</b>
<b>1 Inleiding</b>	<b>1</b>
1.1 Probleemstelling	1
1.2 Onderzoeksvraag	1
1.3 Onderzoeksdoelstelling	2
1.4 Opzet van deze bachelorproef	2
<b>2 Stand van zaken</b>	<b>3</b>
2.1 Artificiële intelligentie (AI)	3
2.1.1 Geschiedenis	4
2.2 Natural Language Processing (NLP)	4
2.2.1 Natural Language Understanding (NLU)	5
2.2.2 Natural Language Generation (NLG)	5
2.3 Transformers	6
2.3.1 Embeddings	6
2.3.2 Zelfaandachtsmechanisme	7
2.3.3 Encoder	7
2.3.4 Decoder	7
2.3.5 Architectuur	7
2.4 Large Language Models (LLMs)	8
2.4.1 GPT	8
2.5 Chatbots	9
2.6 Hybride chatbot	10
2.7 Semantic Kernel	11
2.7.1 Planner	11
2.7.2 Semantische functies	12
2.7.3 Memory	12
<b>3 Methodologie</b>	<b>15</b>
<b>4 Conclusie</b>	<b>16</b>
<b>A Onderzoeksvoorstel</b>	<b>18</b>
A.1 Samenvatting	18
A.2 Introductie	18
A.3 Doelstellingen	19

A.4	Stand van zaken . . . . .	19
A.5	Methodologie . . . . .	20
A.6	Verwachte resultaten . . . . .	20
A.7	Referenties . . . . .	20
<b>Bibliografie</b>		<b>22</b>

# Lijst van figuren

2.1	Mind map van Artificiële Intelligentie . . . . .	4
2.2	Transformer model . . . . .	8
2.3	Verschillende soorten chatbots . . . . .	10
2.4	Het verloop van een vraag met semantic kernel. . . . .	12
2.5	Representatie van een vector databank . . . . .	13

# 1

## Inleiding

Het onderzoeks-domein van artificiële intelligentie is in de laatste jaren op een stroomversnelling geraakt en is in korte tijd voor iedereen met internet toegang en een computer toegankelijk geworden. Deze ontwikkeling is dan ook duidelijk te zichtbaar op de bedrijfswereld zoals blijkt uit de resultaten van een enquête over het gebruik van ICT en e-commerce in bedrijven uitgevoerd door Statbel, het Belgische statistiekbureau. Uit deze enquête blijkt dat bijna een op de zeven bedrijven gebruik maakt van artificiële intelligentie. Bij de bedrijven die meer dan 250 werknemers hebben komt dit aan een op de twee bedrijven. (STATBEL, [2023](#)) Deze bachelorproef kijkt naar het ontwikkelen van een chatbot die kan antwoorden op vragen die te maken hebben met losse data die op verschillende webapplicaties staan.

### 1.1. Probleemstelling

Itineris bied software aan voor je nutsvoorzieningen zoals water of elektriciteit te beheren. Ze bieden hier ook customer service voor aan maar vaak als er een vraag gesteld wordt of er een probleem wordt vastgesteld moeten de medewerkers op meerdere webpagina's of webapplicaties de data gaan bekijken/vergelijken om vast te stellen wat er aan de hand is. Dit neemt tijd in beslag en ze kunnen soms dingen over het hoofd zien.

### 1.2. Onderzoeksvraag

Hoe kunnen we de efficiënte ontwikkeling van een co-pilot bevorderen door het identificeren van verschillende valkuilen en technologieën met specifieke aandacht voor de vereisten van de onderneming?



### 1.3. Onderzoeksdoelstelling

Het onderzoek moet een oog geven op de ontwikkeling van een chatbot en welke technologieën hierbij te pas komen. Een proof of concept wordt hierbij opgesteld om de keuze van de technologieën toe te lichten.

### 1.4. Opzet van deze bachelorproef

De rest van deze bachelorproef is als volgt opgebouwd:

In Hoofdstuk 2 wordt een overzicht gegeven van de stand van zaken binnen het onderzoeksdomein, op basis van een literatuurstudie.

In Hoofdstuk 3 wordt de methodologie toegelicht en worden de gebruikte onderzoekstechnieken besproken om een antwoord te kunnen formuleren op de onderzoeksvragen.

In Hoofdstuk 4, tenslotte, wordt de conclusie gegeven en een antwoord geformuleerd op de onderzoeksvragen. Daarbij wordt ook een aanzet gegeven voor toekomstig onderzoek binnen dit domein.

# 2

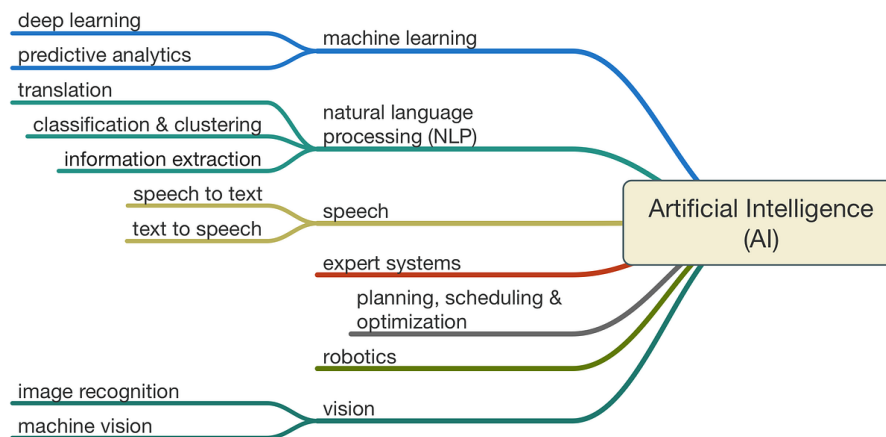
## Stand van zaken

Om te weten welke technologieën we van de artificiële intelligentie kunnen gebruiken voor onze chatbot moeten we eerst het begrip artificiële technologie en toepasselijke technologieën onder deze wetenschap bekijken. Zo is artificiële intelligentie simpel gezegd een algoritme dat intelligentie probeert na te bootsen om bepaalde taken uit te voeren. De effectieve implementaties van artificiële intelligentie kunnen op vele verschillende manieren voorkomen, de nodige gaan we hieronder overlopen.

### 2.1. Artificiële intelligentie (AI)

Artificiële intelligentie, of AI, is technologie dat toelaat om met computers en machines probleem oplossend te denken en menselijke intelligentie na te bootsen (IBM, [z.d.-a](#)). Artificiële intelligentie heeft veel toepassingen en daardoor is het ook een onderwerp dat we steeds vaker verwerken. Dit kan op vele manieren gebeuren en eigenlijk is dit een koepelterm voor meerdere specifieke technologieën (zie onderstaande mind map op figuur 2.1). De verschillende sub-onderwerpen en bepaalde toepassingen hiervan zijn:

- Machine Learning (ML): Wordt gebruikt bij toepassingen zoals zelfrijdende auto's.
- Natural Language Processing (NLP): Wordt gebruikt voor software die met talen werkt zoals Google translate.
- Speech: Wordt gebruikt voor toepassingen zoals Alexa of Siri.
- Expert Systems: Computer programma's die zich kan voordoen als een expert van een vakgebied om oordelen te maken.

**Figuur (2.1)**

Mind map van Artificiële Intelligentie

- Planning, scheduling and optimization: Software om takenvolgorde, toewijzing van middelen en time management.
- Robotics: Het besturen van robots/machines.
- Vision: Beeldherkenning zoals de face recognition op een smartphone.

Deze technologieën staan niet altijd los van elkaar, zo kan het zijn dat de ene technologie gebruik maakt van algoritmes uit een andere technologie.

### 2.1.1. Geschiedenis

Het onderzoeksdomein van artificiële intelligentie is al een stuk langer aanwezig dan wat de meeste mensen denken. Volgens een artikel van [of Europe \(z.d.\)](#) is het nog steeds een relatief “jong” onderzoeksdomein van 60 jaar oud terwijl de meeste mensen er maar in contact mee zijn gekomen de laatste 10 jaren. Geïntantieerd in de tweede wereldoorlog, is de ontwikkeling sterk gelinkt aan die van computers en heeft geleid naar de mogelijkheid om computers steeds ingewikkeldere taken te laten uitvoeren, wat we hiervoor enkel aan mensen konden overlaten ([of Europe, z.d.](#)).

## 2.2. Natural Language Processing (NLP)

Natuurlijke taalverwerking, of NLP, combineert computationele taalkunde (op regels gebaseerde modellering van menselijke taal) met statistische en machine learning-modellen om computers en digitale apparaten in staat te stellen tekst en spraak te herkennen, begrijpen en genereren (IBM, [z.d.-b](#)). Tegenwoordig hebben de meeste mensen hier al contact mee gehad in de vorm van een chatbot, een vertaal assistent of dergelijke. Maar Natural Language Processing speelt ook een steeds belangrijkere rol in bedrijfsoplossingen die helpen om de werklast van de medewerkers te verminderen en de productiviteit een boost te geven. Aangezien Natural Language

Processing instaat voor computers het vermogen te geven van tekst en gesproken taal te begrijpen is het ook deel van wat vaak de drie pillaren van conversationele AI wordt genoemd.

Deze pilaren zijn:

- Natural Language Processing
- Natural Language Understanding
- Natural Language Generation

Deze pilaren hebben elk hun eigen functie en eigen werking maar bouwen allemaal op elkaar om een conversatie tot stand te kunnen brengen. Het weglaten van een van de drie pilaren zou dus zorgen voor een verminderde kwaliteit.

### **2.2.1. Natural Language Understanding (NLU)**

De menselijke taal is heel ingewikkeld en zonder de mogelijkheid voor bepaald stukken tekst te interpreteren kan je soms belangrijke informatie niet bekomen. Natural Language Understanding (NLU) is een cruciaal aspect van de verwerking van natuurlijke taal, waarbij gebruik wordt gemaakt van vooraf getrainde taalmodellen (PLM's) voor representatie (Xu, [2023](#)). NLU omvat het extraheren en weergeven van de betekenis van teksten en dialogen, waarbij ervoor wordt gezorgd dat de interpretatie in lijn is met de bedoelde boodschap van de spreker voor latere redenering en actie bij kunstmatige intelligente agents. Ondanks de vooruitgang blijft het debat over het vermogen van NLU-systemen om de ware taal te begrijpen bestaan, met discussies over de noodzaak van gestructureerde kennisintegratie om interpretatieve en inferentiële vaardigheden te verbeteren. NLU-systemen zijn bedoeld om de kloof tussen computationele modellen en menselijke taalverwerking te overbruggen, waarbij wordt gestreefd naar alomvattend taalbegrip en -generatie binnen intelligente agenten.

### **2.2.2. Natural Language Generation (NLG)**

Eens een computer in staat is van te zien wat er precies in een tekst gezegd is en wat er mee bedoeld wordt zal de computer ook een nuttig antwoord of actie moeten weerleggen. Hiervoor wordt er gebruik gemaakt van Natural Language Generation, NLG is een proces waarbij computersystemen op basis van bepaalde informatie verstaanbare teksten genereren in menselijke taal (Yang & Halim, [2022](#)). NLG heeft aanzienlijke vooruitgang geboekt dankzij Artificiële Intelligentie, waaronder deep-learning-algoritmen en neurale netwerken. NLG omvat het creëren van tekst die bepaalde informatie nauwkeurig weergeeft, grammaticaal correct is en voldoet aan specifieke criteria. Het speelt een rol in verschillende sectoren, zoals business intelligence, klantenservice, gezondheidszorg, onderwijs en cyberbeveiliging, en verbetert de communicatie- en besluitvormingsprocessen (Woo e.a.,

[z.d.](#)). NLG-tools zijn in onderwijsomgevingen gebruikt om de schrijfkwaliteit van studenten te analyseren, wat aantoonde dat meer geavanceerde strategieën in de interactie met NLG-tools leiden tot betere menselijke beoordelingen voor gegenereerde inhoud. Over het algemeen is NLG een essentieel onderdeel van kunstmatige intelligentie, dat het mogelijk maakt om coherente en zinvolle menselijke taal te genereren op basis van gestructureerde gegevens.

## 2.3. Transformers

Transformers zijn geïntroduceerd in 2017 door Google en sinds dan is er een explosie geweest in het aantal geavanceerde modellen (die gerelateerd zijn aan de transformer architectuur) die ontwikkeld werden en het gebruik van deze modellen heeft het belang van Natural Language Processing nog versterkt (Suresh, [2022](#)). Deze modellen hebben de manier veranderd waarop we menselijke taal verwerken en begrijpen, waardoor de grenzen van machinaal leren worden verlegd en opmerkelijke vooruitgang in verschillende NLP-taken mogelijk wordt gemaakt. Van automatische vertaling tot het beantwoorden van vragen en het samenvatten van teksten: Transformers zijn dé architectuur geworden, die ongekennde prestatieniveaus laat zien (Nemeon, [z.d.](#)). Transformer-modellen vertrouwen op zelfaandachtsmechanismen, waardoor ze sneller kunnen leren dan traditionele modellen, zoals modellen voor het lange kortetermijngeheugen (LSTM). Een zelfaandachtig transformatormodel kan naar verschillende delen van een reeks, of de gehele context van een zin, kijken om voorspellingen te genereren.

### 2.3.1. Embeddings

De ruwe tekstdata die we hebben, kan niet worden gebruikt om transformer-modellen te trainen omdat ze enkel getallen begrijpen. Er werden eerder al bewerkingen gebruikt om dingen uit de teksten te halen maar er was er nog geen die de tekst kon coderen in getallen. Embeddings was de eerste manier om begrijpelijke informatie in die getallen te coderen, of de juiste manier van zeggen zou vectoren zijn. Dus in plaats van de tokens naar de encoder door te geven, worden de embeddings ervan doorgegeven. Maar een groot probleem was het coderen van de informatie over de positie van de woorden in de embeddings. Standaard embedding mapte elk token gewoon naar een n-dimensionale vector, maar codeerde nooit de positie-informatie. Om de positie-informatie van de tokens te coderen, kan een andere embedding genaamd positionele embedding worden gebruikt. Bij positionele embedding voeden we in plaats van de tokens van de woorden als invoer de positie-ids of indices van elke zin in. Dit maakt het mogelijk voor de positie-embeddinglaag om een nuttige representatie van de posities te geven. De uiteindelijke uitvoer van de embeddingslaag wordt gegenereerd door de token-embeddings en positie-embeddings toe te voegen en een normalisatie toe te passen op het resultaat zodat er geen enorme waarden zijn (Suresh, [2022](#)).

### 2.3.2. Zelfaandachtsmechanisme

De embeddings die naar de Encoder (zie 2.3.3) worden gestuurd zijn statisch en bevatten een bepaalde hoeveelheid informatie. Het zelfaandachtsmechanisme houdt rekening met de context van de woorden of tokens in een zin en codeert die informatie in hun respectievelijke embeddings. Het coderen van de context in de embeddings is belangrijk omdat homoniemen in een bepaald stuk tekst aan de zelfde embeddings worden toegewezen. Zo zou bijvoorbeeld een bank in de zin van een geldbank dezelfde embeddingwaarde hebben als een bank waarop je gaat zitten in het park. Wanneer de zelfaandachtslaag klaar is met de verkregen embeddings zullen deze woorden hun context gecodeerd hebben. Hiermee kan het model aandachtcores berekenen en op basis hiervan zal het model hogere gewichten toekennen aan tokens die semantisch gerelateerd zijn (een gelijkaardige/gelinkte betekenis hebben) (Suresh, 2022).

### 2.3.3. Encoder

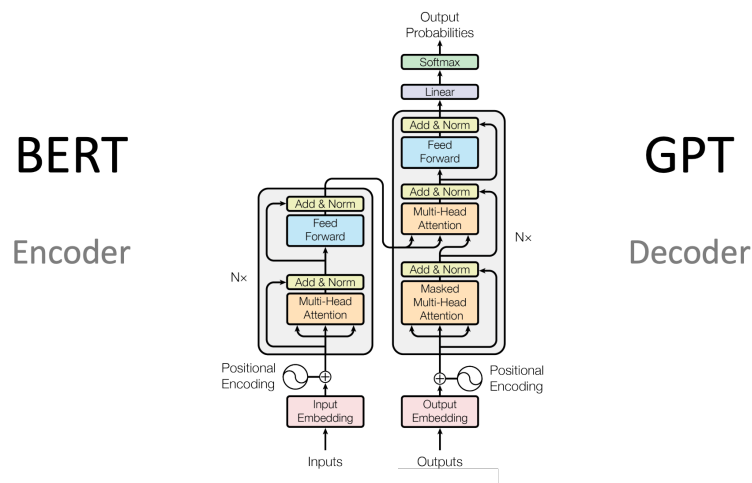
De encoder is verantwoordelijk voor het verwerken van de invoerreeks bestaande uit embeddings en als output zorgt het voor dezelfde vorm als de input maar nu hebben de embeddings veel contextuele informatie zoals betekenis, positie van het woord, etc... (Suresh, 2022). Het bestaat uit een stapel identieke lagen, elke laag bestaat uit een zelfaandachtsmechanisme gevolgd door een feed-forward neurale netwerk. Het zelfaandachtsmechanisme laat het model concentreren op verschillende delen van de invoerreeks en het feed-forward neurale netwerk verwerkt de informatie lokaal. Door deze samenwerking worden niet-lineaire transformaties mogelijk gemaakt.

### 2.3.4. Decoder

De decoder accepteert de gecodeerde representaties van de encoder en genereert een uitvoerreeks. Het bevat ook zelfaandachtsmechanismen om afhankelijkheden binnen de uitvoerreeks vast te leggen en aandacht te besteden aan de relevante delen van de invoerreeks. Hierdoor kan het model contextueel bewuste output genereren voor taken zoals automatische vertaling of samenvatting van teksten.

### 2.3.5. Architectuur

Om de transformator modellen te begrijpen is het soms handig van deze visueel voor te stellen zoals op figuur TODO. In dit voorbeeld zien we een transformator model met als encoder het BERT model en decoder het GPT model. Het BERT model gebruikt het encoder gedeelte van de transformer architectuur om de semantiek van de tekst te begrijpen. Het GPT model gebruikt dan het decoder gedeelte van het transformer model om tekst te genereren voor de output aan de hand van de doorgegeven input van de encoder (Heidloff, 2023).

**Figuur (2.2)**

Transformer model

## 2.4. Large Language Models (LLMs)

Een Large Language Model (LLM) is een deep learning algoritme dat een reeks natuurlijke taalverwerkingstaken kan uitvoeren. De modellen maken gebruik van transformatie modellen en zijn getraind op grote datasets. Zo kunnen ze teksten of andere content herkennen, vertalen, voorspellen of genereren. Naast het aanleren van menselijke talen aan toepassingen van artificiële intelligentie, kunnen large language models ook getraind worden om verschillende taken uit te voeren, zoals het begrijpen van eiwitstructuren, het schrijven van softwarecode en nog veel meer.

### 2.4.1. GPT

Generative Pretrained Transformer modellen zijn taalvoorspellingsmodellen voor algemene doeleinden. Met andere woorden: het zijn computerprogramma's die informatie kunnen analyseren, extraheren, samenvatten en anderszins gebruiken om inhoud te genereren. Generatieve AI technologie is in staat van content te produceren zoals text en fotos waarbij het Pretrained gedeelte betekent dat deze modellen opgeslagen netwerken zijn die al hebben geleerd hoe een bepaalde taak uit te voeren of probleem op te lossen aan de hand van een grote dataset. Deze modellen worden vaak gebruikt in conversationale AI om de User Experience zo persoonlijk en menselijk mogelijk te maken. Er bestaan hier vele modellen onder maar hier zal het vooral over de bekende GPT-3 en GPT-4 van OpenAI gaan, wat ook de modellen zijn die gehanteerd zullen worden in de proof-of-concept. GPT-3 staat voor "Generative Pre-trained Transformer 3", en het is een geavanceerd AI-model dat is ontwikkeld door OpenAI. Het is het derde model in de reeks van "Transformers" die zijn ontworpen voor natuurlijke taalverwerkingstaken. GPT-3 is getraind op een enorme hoeveelheid tekstgegevens en kan vervolgens worden gebruikt voor verschillende taken, zoals het genereren van tekst, het beantwoorden van vragen en

het vertalen van talen. Het model kan opmerkelijk complexe en coherente mensachtige tekst produceren, waardoor het een krachtig instrument is voor een breed scala aan toepassingen, van chatbots tot contentgeneratie (Schulze, 2024). GPT-4 is het meest recente model van OpenAI. Het is een groot multimodaal model (LMM), wat betekent dat zowel beeldinvoer als tekst kan worden geparseerd. Deze iteratie is het meest uitgebreide GPT-model, dat prestaties op menselijk niveau laat zien in verschillende benchmarks op professioneel en academisch gebied. Ter vergelijking: GPT-3.5 behaalt bij de onderste 10 procent van de deelnemers aan een gesimuleerd staafexamen. GPT-4 scoort in de top 10 procent (Schulze, 2024).

## 2.5. Chatbots

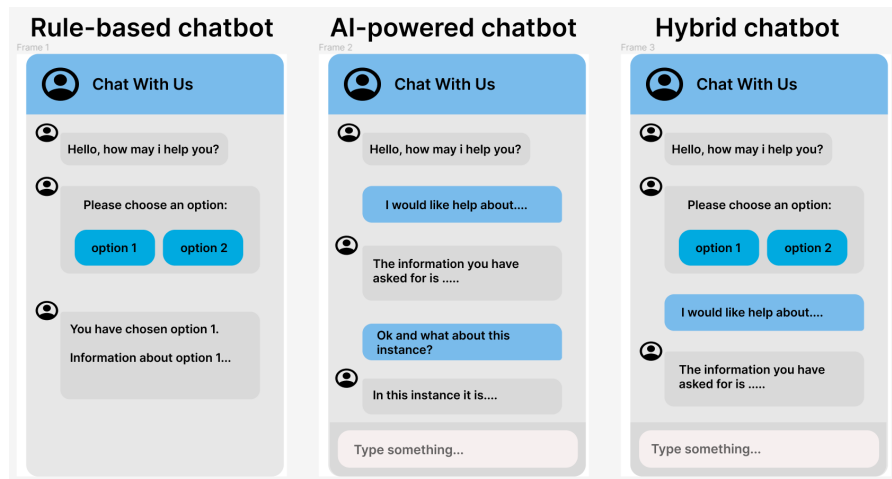
Chatbots bestaan onder vele vormen maar ze hebben allemaal ongeveer hetzelfde doel. Simpelweg gezegd is een chatbot een programma dat conversaties tussen mensen nabootst en verwerkt. Dit laat mensen toe om te intrageren met digitale toestellen zoals ze met een echte persoon zouden doen. Deze programma's kunnen heel simpel zijn om simpele korte antwoorden te geven maar ze kunnen ook heel uitgebreid ontworpen worden dat ze kunnen leren en verbeteren om steeds betere resultaten weer te geven (oracle, z.d.).

Patel (2024) bespreekt enkele voorbeelden van soorten chatbots die gebruikt kunnen worden in de business wereld:

1. **Rule-Based Chatbots:** gebruiken vooraf gedefinieerde regels om specifieke problemen aan te pakken en gesprekken te begeleiden als een flowchart, geschikt voor taken zoals het reserveren van een tafel of het kopen van tickets.
2. **Conversationale AI Chatbots:** Door Machine Learning (ML) en Natural Language Processing (NLP) te combineren, begrijpen deze bots de context en de intentie om reacties te genereren. Ze verbeteren met meer gebruik en bieden gepersonaliseerde ervaringen.
3. **Contextuele Chatbots:** Door gebruik te maken van ML en AI, onthouden deze bots eerdere gesprekken om in de loop van de tijd te leren en zich aan te passen. Voorbeelden zijn Google Assistent, Siri en Alexa.
4. **Hybride Chatbots:** Deze chatbots zijn een combinatie van de simpele Rule-based chatbots en de slimme (conversationale AI) chatbots. Hierdoor kan je het beste van beide werelden verkrijgen door de technologieën te combineren.

In de proof-of-concept die zal ontworpen worden zullen we gebruik maken van een Hybride chatbot. De mogelijkheden om snel veel voorkomende vragen te stellen aan de hand van knoppen zal gecombineerd worden met de conversationale kracht van de Large Language Models om specifieke vragen te stellen. De keuze



**Figuur (2.3)**

Verschillende soorten chatbots

van de Hybride chatbot komt doordat de meeste vragen die binnen komen op de customer service gaan over dezelfde onderwerpen. Aan de hand van sneltoetsen zal het dan mogelijk zijn om deze antwoorden op te halen. Indien er verdere vragen zijn kunnen deze gesteld worden aan de chatbot zelf zoals een conversationele chatbot.

## 2.6. Hybride chatbot

Het bouwen van een hybride chatbot in 2023 omvat het combineren van zowel Rule-Based als machine learning gebaseerde benaderingen om een veelzijdig en effectief conversationeel AI-systeem te creëren. Deze chatbots combineren doorgaans Rule-Based systemen met componenten voor machinaal leren en kunstmatige intelligentie om zo een gepersonaliseerd en menselijkere ervaring te creëren. Het doel is om de sterke punten van beide benaderingen te benutten om het beste van beide werelden te bekomen op de applicatie.

De kernfactoren en belangrijkste attributen van een hybride chatbot zijn (Team, 2023):

- **Rule-Based componenten:** Deze systemen baseren zich op voorafbepaalde regels en beslissingsstructuren. Ze zijn goed op specifieke opdrachten uit te voeren die een gebruiker bijvoorbeeld meer zou nodig hebben. Een voorbeeld hiervan is enkele vragen die een gebruiker vaak heeft die hij aan de hand van een knop kan vragen in plaats van de vraag volledig uit te typen.
- **Machine Learning (ML) en Natural Language Processing (NLP):** Machine Learning en Natural Language Processing bestandsdelen geven de chatbot de kracht om menselijke taal te begrijpen en te genereren. Ze hebben het vermogen om de bedoeling van een gebruiker te begrijpen, entiteiten uit een tekst te halen en context te begrijpen.

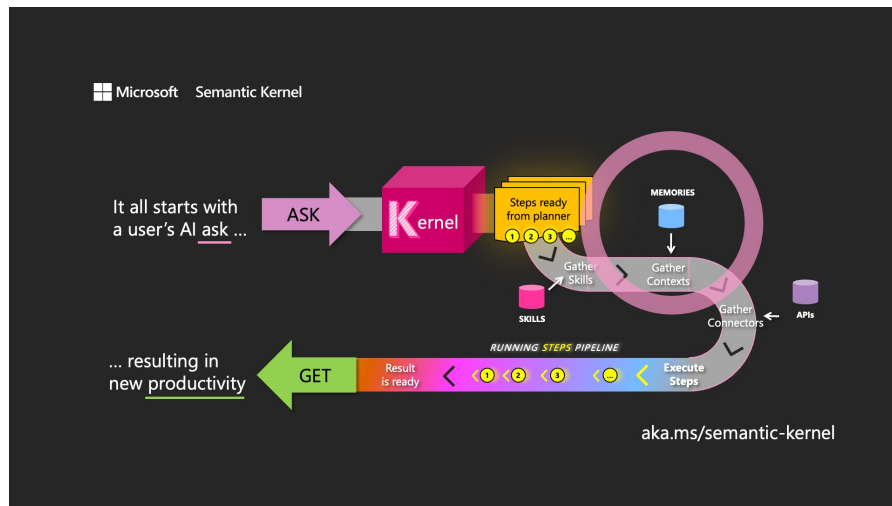
- **Intent Recognition:** Intent Recognition is belangrijk om de chatbot het vermogen te geven van het doel van de gebruiker te ontcijferen. Bijvoorbeeld bij de vraag: Wat is het weer morgen? Zal Intent Recognition het doel identificeren als de weersvoorspelling op te halen.
- **Entity Recognition:** Entity Recognition geeft de chatbot het vermogen van entiteiten/precieze informatie uit de tekst te halen. In het voorbeeld van Intent Recognition over het weer zal het eruit kunnen halen dat de datum 'morgen' is en eventueel een locatie als die wordt meegegeven.
- **Fallback Mechanisms:** Fallback Mechanismen zijn er eigenlijk als soort van vangnetten voor de chatbot als die een bepaalde query/vraag niet kan behandelen. In dit geval zouden de Fallback Mechanismen er dan voor zorgen dat de chatbot een antwoord kan geven door voorafbepaalde antwoorden op vragen die hij niet kent of kan hij extra vragen stellen om meer informatie te verzamelen.

## 2.7. Semantic Kernel

Semantic Kernel (SK) is een AI Software Development Kit (SDK) van Microsoft die u de grote taalmogelijkheden van AI-diensten zoals OpenAI naar uw apps brengt (Salim, [z.d.](#)). Het is een open source development kit met nog steeds actieve updates voor verbetering en verschillende programmeertalen te hanteren. Het wordt in de markt gebruikt om applicaties te maken met aan de basis generatieve AI. Developers kunnen aan de hand van deze development kit gemakkelijk de kracht van de AI services te integreren in hun bestaande applicaties. De semantic kernel staat vooral gekend voor zijn vermogen van plannen op te stellen aan de hand waarvan de applicaties dan handelen. Het is in staat van 'plannen' op te stellen om complexe taken uit te voeren zonder de nodige stappen op voorhand al vastgesteld te hebben. Voor complexere applicaties met meerdere prompts voor bepaalde functies zorgt de semantic kernel voor een een programmatie laag aan de hand van semantische functies. Dit zorgt voor een makkelijke scheiding van je C of python code en je prompts.

### 2.7.1. Planner

De Semantic kernel is een heel handige development kit om je applicatie de kracht van de moderne AI modellen te geven en heeft de kracht om zelf de volgorde van de functies te orkestreren. Daarvoor maakt de semantic kernel gebruik van Planners. Een planner is simpelweg een prompt dat zich baseert op de omschrijving van semantische functies om te kiezen welke nodig zijn en in welke volgorde. Deze planners moeten tegenwoordig bijna altijd gebruik maken van het GPT-4 model om goede plannen op te stellen, andere modellen hebben soms moeite met de juiste semantische functies te kiezen (Salim, [z.d.](#)).

**Figuur (2.4)**

Het verloop van een vraag met semantic kernel.

### 2.7.2. Semantische functies

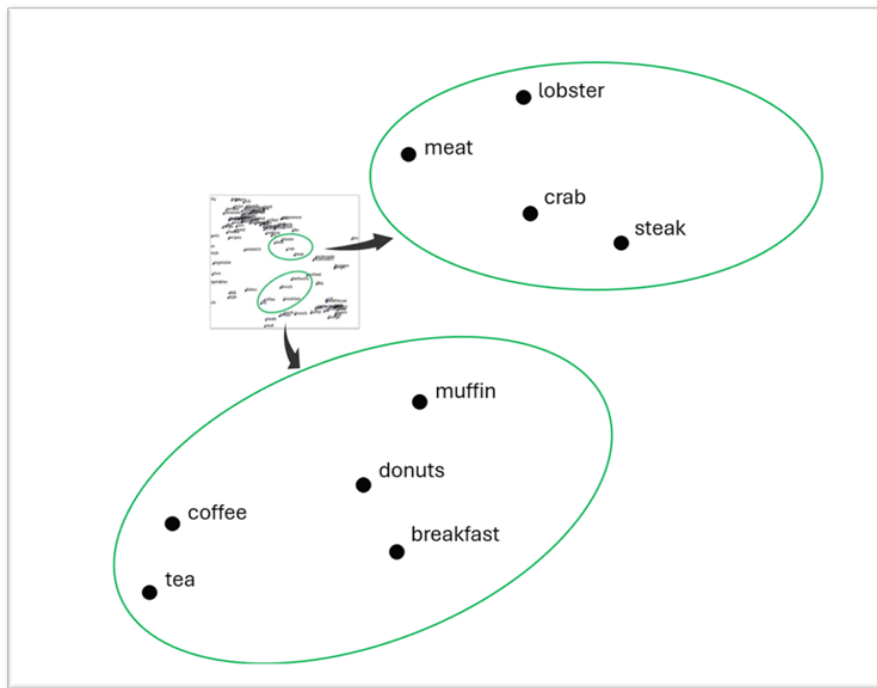
Semantische functies zijn de manier van Semantic Kernel om de LLM prompts na te bootsen. Het grote verschil tussen de LLM prompts en de semantische functies zit hem namelijk in de manier waarop we ze gebruiken. Voor de LLM prompts wordt er vooral gezegd wat er verwacht wordt in grote lijnen en hopen ze dan op de kracht van de LLM om een relevant antwoord te geven, waarbij de semantische functies veel meer in detail over een klein taakje zullen gaan om zo de functies aan elkaar te kunnen koppelen voor een complexere uitkomst gemakkelijk en correct te bekomen. Je kan het dus veel beter sturen van wat je precies als de uitkomst zou willen (Salim, [z.d.](#)).

### 2.7.3. Memory

Een van de meest voorkomende scenario's bij het bouwen van een AI applicatie is dat je de LLM's moet laten werken met je privé data. De LLM's zijn getraind op grote hoeveelheden publieke data en zullen dus voor taken als een onderwerp van een boek te benoemen of een gebeurtenis in de geschiedenis uitleggen gemakkelijk een antwoord kunnen geven. Bij vragen over privé data zoals een ticket van een klant ophalen zal er wat meer bij moeten komen kijken om de LLM's in staat te stellen van een antwoord te formuleren uit de data.

### Retrieval Augmented Generation (RAG)

Een veel voorkomende techniek om de mogelijkheid te geven van met privé data te werken is Retrieval Augmented Generation of RAG. Deze techniek is gebaseerd op het combineren van een Search Engine met de LLM's. In deze techniek wordt de LLM gebruikt om het antwoord te genereren terwijl de Search Engine gebruikt wordt om de relevante info uit de privé data te halen (Cantina, [2024](#)). Het hoofddeel van RAG is dus eigenlijk de zoekfunctie. We kunnen ervan uitgaan dat de LLM's

**Figuur (2.5)**

Representatie van een vector databank

goed zijn om menselijke taal te genereren maar dit hangt bij deze techniek heel hard af van welke resultaten de zoekfunctie kan opleveren. Dit wil dus zeggen dat als de zoekfunctie niet goed werkt of de verkeerde info opbrengt zal het antwoord van de LLM ook niet goed zijn.

### Vector Databases

Door het probleem dat de zoekfunctie kan ondervinden is er een technologie gevonden die vaak in combinatie kan worden gebruikt met RAG: vector databanken of vector indexen. Het idee van vector databanken is dat ze documenten kunnen opslaan en ophalen aan de hand van de gelijkenis van een vraag. Een van de limitaties van het traditioneel zoeken is dat het gebaseerd is op een sleutelwoord en of het woord aanwezig is in het document of titel. Bij vector databanken kan er gezocht worden op de semantiek van een woord, dit wil zeggen dat er documenten kunnen opgehaald worden dat een samenhangende semantiek hebben met de vraag, ook al bevatten ze niet dezelfde woorden. Je kan een vector databank voorstellen als een multidimensionale ruimte waarbij elk document een punt is. De afstand tussen 2 punten is een meetstaaf van hun gelijkaardigheid. Hoe dichter ze bij elkaar liggen hoe meer ze gerelateerd zijn (Cantina, 2024).

In figuur 2.5 kan je een representatie zien van hoe een vector database opgesteld zou zijn. De punten zijn geen fysieke plaatsen in een databank of computer maar deze punten zijn berekend aan de hand van een wiskundige formule. Zo zie je dat in de ene cirkel een verzameling staat van onderwerpen die te maken hebben met

ontbijt en in de andere cirkel gaat het over een grote maaltijd.

### Kernel Memory

Kernel Memory gebruikt veel concepten die RAG gebruikt maar dankzij vele extensie functies kan je het gemakkelijk aansluiten aan veelvoorkomende AI diensten (zoals OpenAI en Azure OpenAI) en vector databanken (zoals Azure AI Search). Het wisselen tussen diensten bij Kernel Memory zou geen aanpassing mogen zijn in de code waardoor je dus niet gebonden bent aan een specifieke dienst als developer. Kernel Memory is ook in staat van meerdere types content in embeddings om te zetten. De lange flow om iets op te halen zoals het in de RAG gedaan zou worden is bij Kernel Memory beschikbaar aan de hand van een enkele methode wat de complexiteit veel verminderd (Cantina, [2024](#)).

Kernel Memory kan op twee manieren werken:

- **Serverless:** De hele dienst wordt rechtstreeks via de applicatie gehost. Dit wil zeggen dat de embeddings worden gegenereerd, opgeslagen en dat de zoekopdracht wordt uitgevoerd binnen de applicatie zelf. Dit is het simpelste scenario maar is niet optimaal voor grote datasets of complexe applicaties doordat het moeilijk te schalen is.
- **As a service:** De service is gehost als een aparte instantie die de applicatie kan gebruiken aan de hand van een REST API. Dit wordt gezien als de aanbevolen aanpak voor complexe applicaties doordat het je toelaat de dienst onafhankelijk te schalen van de hoofdapplicatie. Zo kan je dit bijvoorbeeld hosten op Azure door gebruik te maken van een App Service om het makkelijk te schalen naar meerdere instanties.

# 3

## Methodologie

# 4

## Conclusie

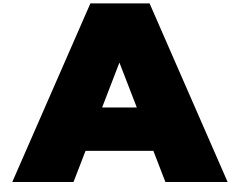
Curabitur nunc magna, posuere eget, venenatis eu, vehicula ac, velit. Aenean ornare, massa a accumsan pulvinar, quam lorem laoreet purus, eu sodales magna risus molestie lorem. Nunc erat velit, hendrerit quis, malesuada ut, aliquam vitae, wisi. Sed posuere. Suspendisse ipsum arcu, scelerisque nec, aliquam eu, molestie tincidunt, justo. Phasellus iaculis. Sed posuere lorem non ipsum. Pellentesque dapibus. Suspendisse quam libero, laoreet a, tincidunt eget, consequat at, est. Nullam ut lectus non enim consequat facilisis. Mauris leo. Quisque pede ligula, auctor vel, pellentesque vel, posuere id, turpis. Cras ipsum sem, cursus et, facilisis ut, tempus euismod, quam. Suspendisse tristique dolor eu orci. Mauris mattis. Aenean semper. Vivamus tortor magna, facilisis id, varius mattis, hendrerit in, justo. Integer purus.

Vivamus adipiscing. Curabitur imperdiet tempus turpis. Vivamus sapien dolor, congue venenatis, euismod eget, porta rhoncus, magna. Proin condimentum pretium enim. Fusce fringilla, libero et venenatis facilisis, eros enim cursus arcu, vitae facilisis odio augue vitae orci. Aliquam varius nibh ut odio. Sed condimentum condimentum nunc. Pellentesque eget massa. Pellentesque quis mauris. Donec ut ligula ac pede pulvinar lobortis. Pellentesque euismod. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent elit. Ut laoreet ornare est. Phasellus gravida vulputate nulla. Donec sit amet arcu ut sem tempor malesuada. Praesent hendrerit augue in urna. Proin enim ante, ornare vel, consequat ut, blandit in, justo. Donec felis elit, dignissim sed, sagittis ut, ullamcorper a, nulla. Aenean pharetra vulputate odio.

Quisque enim. Proin velit neque, tristique eu, eleifend eget, vestibulum nec, lacus. Vivamus odio. Duis odio urna, vehicula in, elementum aliquam, aliquet laoreet, tellus. Sed velit. Sed vel mi ac elit aliquet interdum. Etiam sapien neque, convallis et, aliquet vel, auctor non, arcu. Aliquam suscipit aliquam lectus. Proin tincidunt magna sed wisi. Integer blandit lacus ut lorem. Sed luctus justo sed enim.

Morbi malesuada hendrerit dui. Nunc mauris leo, dapibus sit amet, vestibulum et, commodo id, est. Pellentesque purus. Pellentesque tristique, nunc ac pulvinar adipiscing, justo eros consequat lectus, sit amet posuere lectus neque vel augue. Cras consectetur libero ac eros. Ut eget massa. Fusce sit amet enim eleifend sem dictum auctor. In eget risus luctus wisi convallis pulvinar. Vivamus sapien risus, tempor in, viverra in, aliquet pellentesque, eros. Aliquam euismod libero a sem. Nunc velit augue, scelerisque dignissim, lobortis et, aliquam in, risus. In eu eros. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Curabitur vulputate elit viverra augue. Mauris fringilla, tortor sit amet malesuada mollis, sapien mi dapibus odio, ac imperdiet ligula enim eget nisl. Quisque vitae pede a pede aliquet suscipit. Phasellus tellus pede, viverra vestibulum, gravida id, laoreet in, justo. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Integer commodo luctus lectus. Mauris justo. Duis varius eros. Sed quam. Cras lacus eros, rutrum eget, varius quis, convallis iaculis, velit. Mauris imperdiet, metus at tristique venenatis, purus neque pellentesque mauris, a ultrices elit lacus nec tortor. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent malesuada. Nam lacus lectus, auctor sit amet, malesuada vel, elementum eget, metus. Duis neque pede, facilisis eget, egestas elementum, nonummy id, neque.





# Onderzoeksvoorstel

Het onderwerp van deze bachelorproef is gebaseerd op een onderzoeksvoorstel dat vooraf werd beoordeeld door de promotor. Dat voorstel is opgenomen in deze bijlage.

## **A.1. Samenvatting**

Dit onderzoeksvoorstel richt zich op het ontwikkelen van een co-pilot op een platform dat gebruik maakt van het MS-Dynamics 365 ERP-pakket. Na een voorstelling van het thema en een kadering van de relevantie van copilots in de business omgeving identificeren we de nood aan copilots en de manier om deze te implementeren binnen de applicaties. De onderzoeksvraag richt zich op het identificeren van mogelijke valkuilen die de ontwikkeling van de copilot zou kunnen hinderen en de trainingsmethoden die behandeld zullen worden. De doelstellingen omvatten het identificeren van de belangrijkste vereisten voor de co-pilot, het identificeren van valkuilen die kunnen voorkomen bij de ontwikkeling van de copilot, het trainen van de copilot en een werkend prototype ontwikkelen op basis van het onderzoek. De meerwaarde van dit onderzoek ligt in het bieden van praktische inzichten voor ontwikkelaars en bedrijven die een copilot willen hanteren in hun applicatie. Dit zal de doelgroep betere en snellere hulp bieden bij het ontwikkelen van een copilot.

## **A.2. Introductie**

Artificiële Intelligentie is de wetenschap van de toekomst. Indien je als bedrijf niet op inzet in het jaar 2024 zal je bedrijf een achterstand opbouwen op de competitie. Dit onderzoeksvoorstel streeft naar praktische ondersteuning en info te geven over hoe een co-pilot systeem gebruikt kan worden binnen de bestaande applicatie die Itineris aanbied.

De doelgroep van deze bachelorproef zijn bedrijven die applicaties maken die gebaseerd zijn op het MS Dynamics 365 ERP-pakket.

De onderzoeksvraag luidt dus: **onderzoeksvraag: Hoe kunnen we de efficiënte ontwikkeling van een co-pilot bevorderen door het identificeren van verschillende valkuilen en trainingsmethoden met specifieke aandacht voor de vereisten van de onderneming?**

### **A.3. Doelstellingen**

1. Identificeren van de belangrijkste vereisten voor de co-pilot.
  - Wat moet de co-pilot kunnen?
  - Wat zijn de criteria om te beslissen of een handeling succesvol is?
2. Identificeren van de verschillende valkuilen die we kunnen tegenkomen bij het ontwikkelen van de co-pilot.
  - Wat zijn gekende valkuilen bij het ontwikkelen van een co-pilot?
  - Hoe kunnen we deze valkuilen het beste omzeilen?
3. Trainen van de co-pilot.
  - Hoe train je een co-pilot?
  - Waar zit de informatie die de co-pilot moet verwerken?
  - Hoe gaat de co-pilot trainen met deze informatie?
4. Ontwikkelen van een prototype van een co-pilot die voldoet aan de criteria van punt 1.

### **A.4. Stand van zaken**

Terwijl iedereen nog wende aan het idee van generative AI, wat **GenerativeAI2024** omschrijft als een soort Aritificiële Intelligentie dat een grote verscheidenheid aan data kan genereren zoals fotos, videos, tekst, etc..., is er een nieuwe term opgedoken: AI Copilots.

Een AI co-pilot is in essentie een soort virtuele assistent dat real-time hulp en feedback biedt op handelingen. Het is een "intelligent" programma dat je helpt bij je professionele taken met de nadruk op de mogelijkheid om directe hulp te bieden (**SorabGhaswalla2023**).

Deze co-pilots zijn in staat van te helpen bij het brainstormen en genereren van ideeën, aanpassen en nakijken van uw werk, onderzoeken en factchecken van concepten waarbij het een bepaald niveau van zekerheid kan verzekeren.

Enkele veelgebruikte co-pilots vandaag de dag zijn:

1. Github copilot:

- Een geavanceerde AI assistent die code development voor developers versnelt en de werkdruk verlicht.
2. Microsoft 365 copilot:
    - Een AI assistent die gebruikers helpt bij het uitvoeren van cognitieve taken zoals het schrijven van overtuigende verkooppraatjes en het produceren van presentatiewaardige afbeeldingen.

## **A.5. Methodologie**

1. Literatuurstudie:
  - Onderzoek naar bestaande literatuur en documentatie met betrekking tot de ontwikkeling van co-pilots.
2. Casestudy's:
  - Onderzoek voeren naar bestaande copilots om inzicht te krijgen in de gehanteerde methodologieën en technologieën.
3. Prototype-ontwikkeling:
  - Ontwikkeling van een prototype van een copilot, gebruikmakend van de verzamelde informatie uit de voorafgaande studies.
4. Analyse:
  - Analyseren of de copilot de opdrachten correct uitvoert en voldoet aan de criteria.

## **A.6. Verwachte resultaten**

1. Een overzicht van de belangrijkste vereisten voor schaalbare enterprise webapplicaties.
2. Identificatie van best practices in de ontwikkeling van schaalbare webapplicaties en hun relevantie voor ondernemingen.
3. Een vergelijkende analyse van web development frameworks en technologieën op het gebied van schaalbaarheid en prestaties.
4. Een werkend prototype van een schaalbare enterprise webapplicatie.

## **A.7. Referenties**

<https://generativeai.net/> <https://sorabg.medium.com/who-or-what-is-an-ai-copilot-845175f25ddb>

# Inhoudsopgave

# Bibliografie

- Cantina, T. D. (2024, januari 3). *Build a custom Copilot experience with your private data using and Kernel Memory*. Verkregen mei 10, 2024, van <https://www.developerscantina.com/p/kernel-memory/>
- Heidloff, N. (2023, februari 24). *Foundation Models, Transformers, BERT and GPT*. <https://heidloff.net/article/foundation-models-transformers-bert-and-gpt/>
- IBM. (z.d.-a). *What is AI?* Verkregen mei 2, 2024, van <https://www.ibm.com/topics/artificial-intelligence>
- IBM. (z.d.-b). *What is natural language processing (NLP)?* Verkregen februari 28, 2024, van <https://www.ibm.com/topics/natural-language-processing>
- Nemeon. (z.d.). *Transformers: natuurlijke taalverwerking revolutioneren*. Verkregen maart 22, 2024, van <https://nemeon.io/nl/2023/12/11/transformers-revolutionizing-natural-language-processing/>
- of Europe, C. (z.d.). *History of artificial intelligence*. Verkregen maart 21, 2024, van <https://www.coe.int/en/web/artificial-intelligence/history-of-ai#:~:text=1940-1960:%20Birth%20of%20AI,of%20machines%20and%20organic%20beings.>
- oracle. (z.d.). *what is a chatbot?* Verkregen maart 26, 2024, van <https://www.oracle.com/be/chatbots/what-is-a-chatbot/>
- Patel, S. (2024, maart 21). *5 Different Types of Chatbots for Business Growth*. Verkregen mei 3, 2024, van <https://www.revechat.com/blog/types-of-chatbot/>
- Salim, A. (z.d.). *Understanding semantic kernel*. Verkregen mei 9, 2024, van [https://valoremreply.com/post/understanding-semantic-kernel/#:~:text=Gen%20AI%20applications.-,Semantic%20Kernel%20\(SK\)%20is%20an%20AI%20Software%20Development%20Kit%20\(subsequent%20announcements%20at%20BUILD%202023.](https://valoremreply.com/post/understanding-semantic-kernel/#:~:text=Gen%20AI%20applications.-,Semantic%20Kernel%20(SK)%20is%20an%20AI%20Software%20Development%20Kit%20(subsequent%20announcements%20at%20BUILD%202023.)
- Schulze, J. (2024, maart 19). *What Is GPT? GPT-3, GPT-4, and More Explained*. Verkregen mei 3, 2024, van <https://www.coursera.org/articles/what-is-gpt>
- STATBEL. (2023, december 7). *Bijna een op de twee grote bedrijven gebruikt artificiële intelligentie*. Verkregen mei 1, 2024, van <https://statbel.fgov.be/nl/nieuws/bijna-een-op-de-twee-grote-bedrijven-gebruikt-artificiele-intelligentie>
- Suresh, S. K. (2022, oktober 12). *Understanding Transformers — Encoder*. Verkregen mei 6, 2024, van <https://medium.com/@mr.sk12112002/understanding-transformers-encoder-1f269blcc943#:~:text=Transformer%20Architecture:>

&text=The%20encoder%20takes%20in%20the,,%20etc-encoded%20in%20them.

Team, C. (2023, oktober 3). *Hybrid Chatbot: Everything You Need To Know (2023 Complete Guide)*. Verkregen mei 9, 2024, van <https://www.linkedin.com/pulse/hybrid-chatbot-everything-you-need-know-2023-complete-guide>

Woo, D. J., Susanto, H., & Wang, Y. (z.d.). Natural Language Generation in Student Writing: Language Features, Strategies and Links to Successful Writing.

Xu, L. (2023, juni 5). *Enhancing Language Representation with Constructional Information for Natural Language Understanding*.

Yang, A., & Halim, S. (2022). Natural Language Generation Using Machine Learning-Techniques. *Journal of Student Research*.