

De digitale helpdesk voor verzekervragen

Een onderzoek naar de werking van IBM Watson als digitale helpdesk binnen de verzekeringsbranche, en de implementatie van een prototype.

Versie 1.4

Mei, 2016

Auteur	Rick Koster
Studentnummer	1622528
Opleiding	Hbo-ICT
Studierichting	Technische Informatica
Onderwijsinstelling	Hogeschool Utrecht
Afstudeerbedrijf	Colimbra B.V.
Docentbegeleider	Henk van Nimwegen
Bedrijfsbegeleider	Ronald van den Berg
Eerste examiner	Joop Kaldeway
Tweede examiner	Henk van Nimwegen

Hbo-bachelor scriptie

Hogeschool Utrecht | Colimbra B.V. | IBM Watson

De digitale helpdesk voor verzekeringsvragen

Een onderzoek naar de werking van IBM Watson als digitale helpdesk binnen de verzekeringsbranche, en de implementatie van een prototype.

Versie 1.4
Mei, 2016

Auteur**Rick Koster****S:** 1622528**E:** rick.koster@student.hu.nl**E:** rkoster@colimbra.comOnderwijsinstelling**Hogeschool Utrecht**

Nijenoord 1

3552 AS Utrecht

T: 088 481 82 83**E:** info@hu.nlAfstudeerbedrijf**Colimbra B.V.**

Kruisweg 821

2132 NG Hoofddorp

T: 023 565 37 38**E:** info@colimbra.com**Docentbegeleider**
Bedrijfsbegeleider

Henk van Nimwegen

Ronald van den Berg

Eerste examiner**Tweede examiner**

Joop Kaldeway

Henk van Nimwegen

Legenda**S:** Studentnummer**T:** Telefoonnummer**E:** E-mailadres

Management samenvatting

Dit document is opgesteld in opdracht van Hogeschool Utrecht als onderdeel van het afstuderen. Hierin wordt teruggeblikt op de afstudeerperiode van Rick Koster, 4^{de} jaar student Technische Informatica. De afstudeeropdracht is uitgevoerd bij het bedrijf Colimbra B.V. gedurende de periode van februari 2016 tot en met juni 2016.

Coimbra B.V. is een IT-partner die software ontwikkelt, beheert, implementeert en onderhoudt binnen de verzekeringsbranche. Eind 2015 is Coimbra partner geworden van het IBM Watson ecosysteem om een implementatie van IBM Watson te kunnen realiseren.

Binnen dit document staat de volgende onderzoeksvraag centraal:

Op welke manier kunnen vragen binnen de verzekeringsbranche, in natuurlijke taal, beantwoord worden met behulp van IBM Watson en hoe kan de werking hiervan bewezen worden met een prototype (proof of concept)?

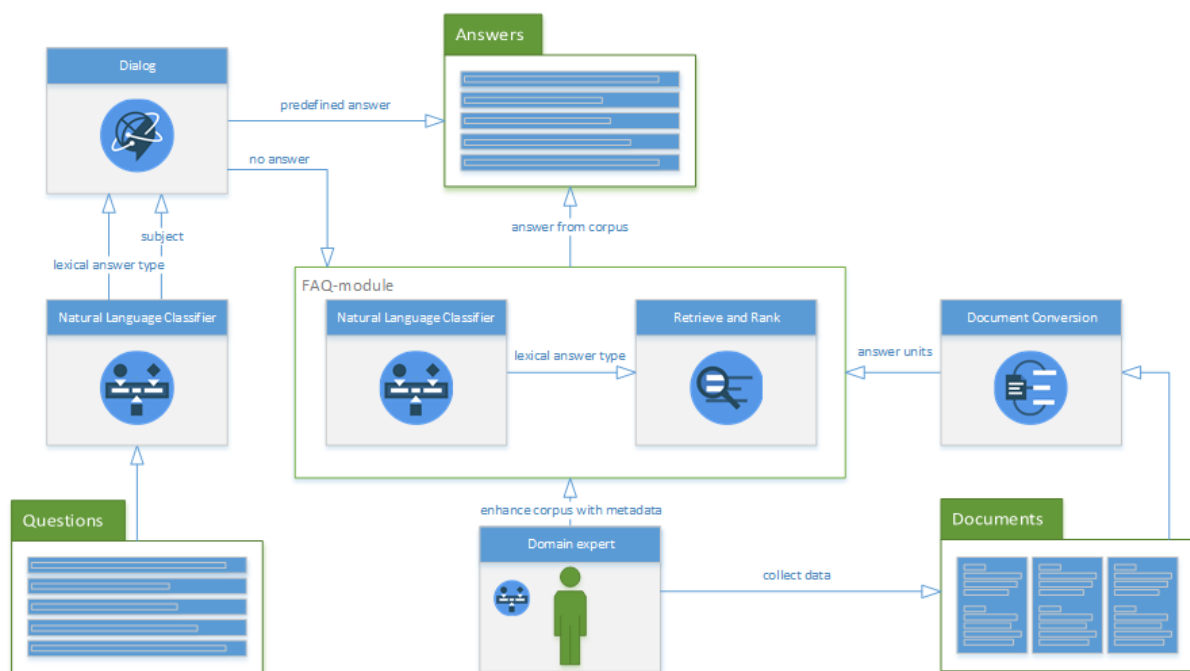
Conclusie

Om deze vraag te beantwoorden is gekeken naar de beschikbare Watson services en welke van deze services relevant zijn om het gewenste prototype te realiseren.

Vragen binnen de verzekeringsbranche kunnen beantwoord worden met behulp van Watson door in ieder geval gebruik te maken van de volgende Watson services:

- Dialog service;
- Document Conversion service;
- Natural Language Classifier service;
- Retrieve and Rank service.

De manier waarop een vraag- en antwoordsysteem opgesteld kan worden met deze services is door de onderstaande architectuur te gebruiken. Meerdere informatiebronnen kunnen toegevoegd worden door meerdere modules te gebruiken. Bijvoorbeeld door gebruik te maken van een module met polisvoorwaardendocumenten en een module om Wikipedia te doorzoeken.



De Document Conversion service wordt gebruikt in combinatie met de Natural Language Classfier service om een corpus op te bouwen. Deze corpus wordt ingeladen in de Retrieve and Rank service waarna deze getraind wordt om vragen te kunnen linken naar de ingelezen antwoorden. De Retrieve and Rank service wordt echter alleen gebruikt om niet vooraf gedefinieerde vragen te beantwoorden met behulp van een citaat. Vaak gestelde vragen worden verwerkt in de Dialog service. De manier waarop deze vaak gestelde vragen herkend worden is door de Natural Language Classfier service in te zetten om het verwachte antwoordtype (lexical answer type) en de context van de vraag te achterhalen.

Het prototype waarnaar verwezen wordt in de hoofdvraag zal eind juni 2016 gepresenteerd worden op een verzekeringscongres. Voor de realisatie van dit prototype is het vraag- en antwoordsysteem verwerkt in de bestaande chatapplicatie: "Telegram". Hierdoor zijn verschillende interfaces beschikbaar om met het systeem te communiceren.

Aanbevelingen

Hieronder worden een aantal aanbevelingen opgesomd op functioneel en technisch gebied. Deze aanbevelingen geven aan op welke manier de Watson implementatie uitgebreid kan worden.

Functioneel

Colimbra beschikt over een ontwikkelingssuite, de Colimbra Application Solution (CAS), die intern gebruikt wordt om webapplicaties te ontwikkelen. In de toekomst zal Colimbra waarschijnlijk de Watson implementatie willen verwerken in CAS. Wanneer dit gebeurt zal de Dialog service overbodig worden omdat het aanmaken van de dialoogstructuur ook in CAS kan gebeuren.

Voor het prototype is besloten om een module te gebruiken waarmee vragen die regelmatig gesteld worden beantwoord kunnen worden. Aangezien deze module vaak gestelde vragen beantwoord is het beter om deze te verwerken in de Dialog service. En de modules alleen in te zetten voor vragen die niet vooraf zijn gedefinieerd.

Watson kan beter en met een hogere zekerheidsgraad vragen beantwoorden naarmate meer trainingsdata beschikbaar is. Hierdoor is het handig om voor een volgend prototype een verzekeraar te benaderen die trainingsdata kan leveren. Denk hierbij bijvoorbeeld aan gesprekken van helpdeskmedewerkers of een interne database.

Technisch

Wanneer de Dialog service wordt vervangen door CAS zal de `WatsonInstance` klasse aangepast moeten worden. De `WatsonInstance` klasse implementeert de architectuur van het vraag- en antwoordsysteem. Deze architectuur kan ook gedefinieerd worden in CAS. Hierdoor moet CAS echter uitgebreid worden om de benodigde Watson services aan te kunnen roepen.

Inhoudsopgave

1. Inleiding.....	1
1.1 Doel	1
1.2 Vooruitblik.....	1
2. Colimbra	3
2.1 Bedrijfscontext	3
2.2 Kwestie.....	3
2.3 Aanleiding	4
2.4 Doelstelling	4
3. Theoretisch Kader	5
3.1 Hoofdvraag.....	5
3.2 Voorlopige deelvragen	5
3.3 Vooronderzoek	5
3.4 Definitieve deelvragen	7
4. Opdrachtomschrijving	8
4.1 Beschrijving	8
4.2 Onderzoeksfase.....	8
4.3 Ontwikkelfase	10
5. IBM Watson	11
5.1 International Business Machines (IBM)	11
5.2 Watson	11
5.3 Producten.....	12
5.4 Prototype	12
6. Watson use case	13
6.1 Beschrijving	13
6.2 Prototype	14
7. Watson serviceoverzicht.....	15
7.1 Beschikbare services	15
7.2 Relevante services.....	17

8. Servicebeschrijving.....	18
8.1 Dialog	19
8.2 Document conversion	21
8.3 Natural Language Classifier	22
8.4 Retrieve and Rank	24
8.5 Configuratiemogelijkheden.....	26
9. Vraag- en antwoordsysteem.....	27
9.1 Architectuur	28
9.2 Modules	29
9.3 Prototype	30
10. Overige benodigdheden	31
10.1 Interface.....	32
10.2 Database	32
11. Training.....	33
12. Prototype.....	34
12.1 WatsonAPI.....	35
12.2 WatsonCore	37
12.3 TelegramAPI	39
12.4 ColimbraBot	40
13. Conclusie en aanbevelingen.....	42
13.1 Conclusie	42
13.2 Aanbevelingen	43
14. Procesgang evaluatie.....	44
15. Bibliografie.....	45

16. Bijlagen	46
Bijlage A: Plan van Aanpak	46
Bijlage B: Functioneel ontwerp	46
Bijlage C: Technisch ontwerp	46
Bijlage D: Watson serviceoverzicht	46
Bijlage E: Prototype	46

Figurenlijst

Figuur 1 Implementatie voorbeeld	10
Figuur 2 Use case diagram	13
Figuur 3 MoSCoW methode	14
Figuur 4 Dialog service - implementatie workflow (IBM, 2016d)	19
Figuur 5 Document conversion service - implementatie workflow (IBM, 2016d)	21
Figuur 6 Natural language classifier service - implementatie workflow (IBM, 2016d)	22
Figuur 7 Natural Language Classifier service - groundtruth voorbeeld.	23
Figuur 8 Retrieve and rank service - implementatie workflow (IBM, 2016d)	24
Figuur 9 Retrieve and Rank service - voorbeeld Solr-configuratie	25
Figuur 10 Retrieve and Rank service - groundtruth voorbeeld	25
Figuur 11 Architectuurdiagram - kern	28
Figuur 12 Architectuur diagram - uitgebreid	29
Figuur 13 Architectuurdiagram - prototype	30
Figuur 14 Dialog service	35
Figuur 15 Document Conversion service	35
Figuur 16 Natural Language Classifier service	36
Figuur 17 Retrieve and Rank service	36
Figuur 18 Watson service	37
Figuur 19 Watson gebruiker	37
Figuur 20 Watson Core configuratie	38
Figuur 21 TelegramBot service	39
Figuur 22 ColimbraBot service	40
Figuur 23 ColimbraBot commando voorbeeld	40
Figuur 24 ColimbraBot configuratie	41
Figuur 25 Architectuur - conclusie	42

Afkortingen en begrippen

Onderstaand een, op alfabet gesorteerde, opsomming van afkortingen en begrippen die gebruikt zijn binnen dit document. Afkortingen kunnen terugkomen in de begrippenlijst met een korte uitleg.

Afkortingen

AI

Artificial Intelligence (kunstmatige intelligentie)

API

Application Programming Interface

CAS

Colimbra Application Solution

CLR

Common Language Runtime

DI

Dependency Injection

IBM

International Business Machines

POCO

Plain Old CLR Object (standaard dataklasse zonder extra functionaliteit)

REST

Representational State Transfer

Begrippen

Application Programming Interface: Een interface waarmee met behulp van programmeercode een bepaalde applicatie/service benaderd kan worden.

Artificial Intelligence: De Engelse benaming voor kunstmatige intelligentie. Wanneer gesproken wordt over een AI wordt vaak een supercomputer bedoeld met menselijke intelligentie.

Bluemix: Een cloud platform van IBM waar applicaties in gehost kunnen worden. Deze applicaties hebben via Bluemix toegang tot verschillende IBM-services (IBM, 2016a).

Cognitief: Een applicatie is cognitief als het mogelijk is om door ervaringen met gebruikers verder te leren en zichzelf te verbeteren.

Corpus: Onder een corpus wordt een verzameling van "answer units" verstaan. Dit is een JSON-geformatteerde collectie van gegevens waar Watson toegang tot heeft.

Cross-platform: Een applicatie wordt cross-platform genoemd wanneer deze op meerdere besturingssystemen kan draaien (Linux, OSx, Windows).

International Business Machine (IBM): Een Amerikaans bedrijf dat zich bezighoudt met de ontwikkeling van hardware, middleware en software, en het bieden van infrastructuur, hosting en consultancy services. Besproken in hoofdstuk 5.

Jeopardy!: Een Amerikaans tv-programma waarin een quiz wordt gespeeld. Tijdens deze quiz worden antwoorden gegeven waarna de deelnemers de bijbehorende vraag moeten verzinnen.

Module: Voor het eerst besproken in hoofdstuk 9. Een module representeert een bron van data voor de Watson implementatie.

Natuurlijke taal: Een natuurlijke taal of etnische taal is in de taalfilosofie elke levende of dode taal die door een aantal mensen als hun moedertaal wordt of werd beschouwd. Taal waarvan de regels zijn afgeleid uit het gangbare gebruik in plaats van op formele wijze te zijn opgesteld, zoals bij een programmeertaal.

REST-API: Representational state transfer application programming interface. Dit houdt in dat de services die gebruik maken van REST-API te benaderen zijn via webrequests. Deze services worden ook wel RESTful services genoemd (Massé, 2012).

Solr: Een open-source zoekmachine. Dit project maakt het mogelijk om een zoekmachine te verwerken in een applicatie (Apache Software Foundation, 2016).

Watson: Een supercomputer ontwikkeld door International Business Machines (IBM). Deze supercomputer is in staat om antwoorden te geven op vragen die gesteld zijn in natuurlijke taal. Een uitgebreidere beschrijving van IBM Watson is gegeven in hoofdstuk 5.

1. Inleiding

Dit document is opgesteld in opdracht van Hogeschool Utrecht als onderdeel van het afstuderen. Hierin wordt teruggeblikt op de afstudeerperiode van Rick Koster, 4^{de} jaar student Technische Informatica. De afstudeeropdracht is uitgevoerd bij het bedrijf Colimbra B.V. gedurende de periode van februari 2016 tot en met juni 2016.

1.1 Doel

Vanuit de verzekeringsbranche is een grote vraag naar innovatie ontstaan. Het bedrijf Colimbra B.V. (hierna Colimbra genoemd) wil hierop inspelen door IBM Watson (hierna Watson genoemd) in te zetten als digitale helpdesk.

Watson is een supercomputer van het bedrijf International Business Machines (hierna IBM genoemd). Watson begrijpt natuurlijke taal en kan getraind worden om een bepaalde sector, in dit geval de verzekeringsbranche te begrijpen. Documenten die aan Watson worden gegeven zullen omgezet worden in kennis waardoor Watson uiteindelijk in staat is om vragen te beantwoorden.

De implementatie van Watson als digitale helpdesk binnen de verzekeringsbranche is interessant omdat dit voor verzekeraars en hun klanten voor een besparing van kosten zorgt. Voor een verzekeringsvraag moet in de huidige situatie eerst door polisdocumenten en de bijbehorende polisvoorwaarden gezocht worden. Als het antwoord niet gevonden wordt kan gebeld worden naar de verzekeraar die uiteindelijk aan de hand van dezelfde documenten tot een conclusie zal komen. Watson zou ook in staat moeten zijn om met deze documenten een vraag te beantwoorden.

1.2 Vooruitblik

Hieronder volgt een vooruitblik op de hoofdstukken die besproken worden.

Achtergrond

Colimbra

De afstudeeropdracht wordt uitgevoerd voor Colimbra. Colimbra is een IT-partner die software ontwikkelt, beheert, implementeert en onderhoudt binnen de verzekeringsbranche. Een beschrijving van Colimbra wordt gegeven in hoofdstuk 2.

Theoretisch Kader

Nadat duidelijk is wat Colimbra wil bereiken wordt het theoretische kader besproken. Dit gebeurt in hoofdstuk 3.

Opdrachtschrijving

De opdracht bestaat uit het onderzoeken van de werking van Watson om uiteindelijk een prototype te realiseren. Dit prototype zal overigens eind juni 2016 worden gepresenteerd op een verzekeringscongres. De afstudeerperiode is te verdelen in een onderzoeksfase en een ontwikkelfase. Een meer gedetailleerde beschrijving van de opdracht wordt gegeven in hoofdstuk 4.

Onderzoeksfase

IBM Watson

Om de services van Watson te kunnen implementeren moet eerst begrepen worden wat Watson is en waar Watson voor dient. Een beschrijving over Watson en de mogelijkheden van Watson worden gegeven in hoofdstuk 5.

Watson use case

Colimbra wil Watson inzetten als digitale helpdesk binnen de verzekeringsbranche. Om te achterhalen wat verwacht wordt van Watson moet eerst vastgelegd worden over welke functionele eigenschappen het prototype moet beschikken. Deze functionele eigenschappen worden beschreven in hoofdstuk 6.

Watson serviceoverzicht

Watson is niet beschikbaar als product maar als verzameling van services. Er worden op dit moment 18 services aangeboden welke gebruik maken van een, door IBM gehost, Watson instantie. Een overzicht van deze services wordt gegeven in hoofdstuk 7.

Servicebeschrijving

Uit de 18 services die beschreven zijn in hoofdstuk 7 bleken vier services relevant te zijn voor het prototype. Deze services worden in meer detail beschreven in hoofdstuk 8.

Vraag- en antwoordsysteem

De vier losstaande services uit hoofdstuk 8 moeten uiteindelijk met elkaar samenwerken om een functionerend vraag- en antwoord systeem op te zetten. De manier waarop dit gebeurt wordt beschreven in hoofdstuk 9.

Overige benodigdheden

Met behulp van de Watson services is het mogelijk om het gewenste vraag- en antwoordsysteem op te zetten. Het prototype zal echter eind juni 2016 gepresenteerd worden op een verzekeringscongres waardoor een presenteerbare interface gewenst is. Een interface wordt niet aangeboden vanuit IBM en zal zelf geïmplementeerd moeten worden. Ook is de wens om het gebruik van de applicatie te loggen zodat Watson met deze informatie verder getraind kan worden. De oplossingen die hiervoor gekozen zijn worden beschreven in hoofdstuk 10.

Training

Voordat Watson gebruikt kan worden binnen de verzekeringsbranche moet het eerst getraind worden om deze branche te begrijpen. De manier waarop dit gebeurt inclusief de eisen die gesteld worden aan de trainingsdata worden gegeven in hoofdstuk 11.

Ontwikkelfase**Prototype**

Na de onderzoeksfase wordt voordat een conclusie wordt getrokken eerst een prototype ontwikkeld waarmee de werking van Watson getoetst wordt (proof of concept). Het ontwikkelingsproces van dit prototype wordt beschreven in hoofdstuk 12.

Afronding**Conclusie en aanbevelingen**

Met behulp van de voorgaande hoofdstukken kan een conclusie getrokken worden. Deze conclusie inclusief aanbevelingen voor Colimbra wordt gegeven in hoofdstuk 13.

Procesgang evaluatie | Bibliografie | Bijlagen

Een evaluatie op de procesgang wordt gegeven in hoofdstuk 14. Daaropvolgend worden de gebruikte bronnen opgesomd in hoofdstuk 15. Tot slot zullen de bijlagen worden toegevoegd in hoofdstuk 16.

2. Colimbra

Binnen dit hoofdstuk wordt een beschrijving gegeven van Colimbra. De bedrijfscontext wordt beschreven in paragraaf 2.1. De kwestie waaraan wordt gewerkt, wordt beschreven in paragraaf 2.2. De aanleiding tot het uitvoeren van dit project wordt gegeven in paragraaf 2.3 met daarna in paragraaf 2.4 de uiteindelijke doelstelling.

2.1 Bedrijfscontext

Zoals in de inleiding al is aangegeven, zal de afstudeeropdracht uitgevoerd worden voor Colimbra. Colimbra is een IT-partner (Information Technology partner) die software ontwikkelt, beheert, implementeert en onderhoudt binnen de verzekeringsbranche. Colimbra heeft een relatief kleine omvang van ongeveer 20 werknemers waarvan de meeste domeinexperts zijn. Vier van deze medewerkers zijn systeemontwikkelaars die werken aan de Colimbra Application Solution (CAS). De CAS-suite wordt binnen Colimbra gebruikt door acht applicatieontwikkelaars om webapplicaties te realiseren.

Uitvoering van de opdracht zal resulteren in een onderzoeksverslag (dit document), een functioneel ontwerp (bijlage B), een technisch ontwerp (bijlage C) en een losstaand prototype. Deze producten zullen gerealiseerd worden op de ontwikkelafdeling van Colimbra. Uiteindelijk kan deze service resulteren in een toevoeging aan de 'Mijn omgeving' service, een service aangeboden door Colimbra waarmee alle polisgegevens van de uiteindelijke klant op overzichtelijke wijze bij elkaar worden gehouden (Kleeff, 2015).

Colimbra richt zich op drie gebieden: front- en midoffice oplossingen als extranetten en doelgroep omgevingen, koppelvraagstukken voor berichtenverkeer en documentuitwisseling, en klantbedieningsconcepten (Colimbra, 2015).

2.2 Kwestie

Tijdens de afstudeerperiode zal gewerkt worden aan het eenvoudiger maken om vragen te stellen binnen de verzekeringsbranche. Met een verzekeringsvraag is het zo dat door polisdocumenten en de bijbehorende polisvoorwaarden gezocht moet worden om een antwoord te vinden. Als er dan nog onduidelijkheden zijn kan de helpdesk van de desbetreffende verzekeraar worden gebeld. Dit zorgt echter voor extra kosten voor het desbetreffende verzekeringsbedrijf (klant van Colimbra) en de vragende persoon (uiteindelijke klant).

Om deze kwestie te verbeteren wordt dit onderzoek uitgevoerd naar de werking van Watson en de hoeveelheid werk dat verricht moet worden voor de ontwikkeling van een digitale helpdesk. Watson is interessant voor deze kwestie omdat Watson, zoals eerder in de inleiding al is aangegeven, in staat is om natuurlijke taal te begrijpen en daardoor kan daardoor antwoorden op gestelde vragen.

Dit zal het vinden van een antwoord op verzekeringsvragen eenvoudiger maken voor de uiteindelijke klant omdat het niet nodig is om zelf documenten te doorzoeken. Aangezien het een digitale helpdesk betreft is het voor de uiteindelijke klant mogelijk om kosteloos vragen te stellen en kan de drempel om een vraag te stellen lager liggen. Dit omdat het niet nodig is om een helpdeskmedewerker te bellen.

2.3 Aanleiding

Binnen de verzekeringsbranche worden een groot aantal data en documenten verwerkt. Deze data en documenten kunnen zoals eerder al aangegeven met behulp van Watson worden omgezet tot kennis, waardoor via een dialoog gezocht kan worden naar deze gegevens. De aanleiding om deze opdracht uit te voeren is dan ook om het bemachtigen van verzekeringsdata eenvoudiger te maken voor de uiteindelijke klant. Met verzekeringsdata worden hier antwoorden op verzekeringsvragen bedoeld. Door de mogelijke verlaging van vragen die naar de helpdesk gaan kunnen hierdoor ook kosten worden bespaard.

De data waarover gesproken wordt is bijvoorbeeld afkomstig uit een dekkingsoverzicht of een opgestelde polis. Wanneer Watson deze documenten op een juiste manier inleest en begrijpt zullen bijvoorbeeld de volgende vragen beantwoord kunnen worden:

- **Als ik met mijn auto in het buitenland tegen een paaltje aan rijd, wordt mijn schade dan vergoed door de verzekering?**
- **Heb ik een extra verzekering nodig als ik op wintersport ga?**
- **Ik ben laatst getrouwd, welk effect heeft dit op mijn verzekeringsdekking?**
- **Wat zijn de dekkingen voor sieraden?**

Volgens een vooraf opgestelde business-case door (Kleeff, 2015), is ook de bedrijfsstrategie van Colimbra een belangrijke aanleiding voor het oppakken van deze kwestie: "Colimbra wil nieuwe producten aan blijven bieden voor hun bestaande en potentieel nieuwe klanten". Colimbra wil met het oppakken van deze kwestie inspelen op de vraag naar innovatie vanuit de verzekeringsbranche. Dit geeft Colimbra de kans om vernieuwende producten aan te blijven bieden voor hun bestaande en potentieel nieuwe klanten.

2.4 Doelstelling

De situatie waar Colimbra zich in bevindt is dat Watson interessant klinkt door de mogelijkheden van Watson. Dat het project uitvoerbaar is staat vast. Het is echter nog onduidelijk welke functionaliteit de beschikbare tools/services bieden en wat er dus moet gebeuren om een prototype te kunnen ontwikkelen.

Het doel van de opdracht is om het voor klanten van verzekeraars eenvoudiger en goedkoper te maken om vragen te stellen. Dit kan worden gerealiseerd door gebruik te maken van Watson. Om dit doel te bereiken is het dus noodzakelijk om te achterhalen hoe de services van Watson werken en op welke manier deze gebruikt kunnen worden om een vraag- en antwoordsysteem op te zetten.

Zoals in de inleiding is vermeld wordt eind juni 2016 de werking van Watson gepresenteerd op een verzekeringscongres. Het vraag- en antwoordsysteem moet daarom verwerkt worden in een prototype. Dit prototype dient als proof of concept voor de werking van Watson.

3. Theoretisch Kader

Dit hoofdstuk zal het theoretisch kader beschrijven. In paragraaf 3.1 wordt de hoofdvraag gegeven. In het beginstadium zijn hiervoor voorlopige deelvragen opgesteld, deze worden genoemd in paragraaf 3.2. Paragraaf 3.3 beschrijft kort het vooronderzoek, in dit vooronderzoek worden de voorlopige deelvragen verder afgebakend tot definitieve deelvragen. Deze definitieve deelvragen worden uiteindelijk opgesomd in paragraaf 3.4.

3.1 Hoofdvraag

In paragraaf 2.4 is de doelstelling van Colimbra gegeven. Om deze doelstelling te behalen wordt een onderzoek uitgevoerd waarin de onderstaande hoofdvraag centraal staat:

Op welke manier kunnen vragen binnen de verzekeringsbranche, in natuurlijke taal, beantwoord worden met behulp van IBM Watson en hoe kan de werking hiervan bewezen worden met een prototype (proof of concept)?

3.2 Voorlopige deelvragen

Om de hoofdvraag uit paragraaf 3.1 te beantwoorden is deze afgebakend in deelvragen. In eerste instantie zijn de volgende deelvragen opgesteld:

1. Wat is IBM Watson?
2. Op welke manier kan Watson geïmplementeerd worden om documenten uit de verzekeringsbranche te begrijpen?
3. Welke Watson API's zijn voor Colimbra geschikt om te gebruiken, en hoe kunnen deze gebruikt worden?
4. In hoeverre dient IBM Watson getraind te worden om kennis op te doen over de verzekeringsbranche?

Omdat deelvragen 2 en 3 vrij globaal zijn is voor de start van het onderzoek een vooronderzoek uitgevoerd. Dit vooronderzoek is beschreven in paragraaf 3.3.

3.3 Vooronderzoek

De voorlopige deelvragen uit paragraaf 3.2 zijn kort bekeken om concrete deelvragen op te kunnen stellen. Hieronder staat kort omschreven hoe de definitieve deelvragen zijn ontstaan.

IBM Watson

Watson is een supercomputer van IBM die de natuurlijke taal begrijpt en getraind kan worden om een bepaalde sector, in dit geval de verzekeringsbranche, te begrijpen. Dit gebeurt door de mogelijkheid van Watson om documenten of webpagina's om te zetten naar kennis. Uiteindelijk kan Watson getraind worden als "expert" binnen de verzekeringsbranche, wat het onder andere mogelijk maakt om Watson in te zetten als digitale helpdesk.

De eerste implementatie van Watson is ontwikkeld om deel te nemen aan "Jeopardy!", een Amerikaans tv-programma waarin een quiz wordt gespeeld. Tijdens deze quiz worden antwoorden gegeven waarna de deelnemers de bijbehorende vraag moeten verzinnen. Om te winnen moest Watson eerst het antwoord begrijpen en daar daarna de bijbehorende vraag bij formuleren. Watson heeft uiteindelijk gewonnen van de, in die tijd, twee beste spelers die hebben meegespeeld: Ken Jennings en Brad Rutter (Best, 2015).

In een gesprek met (Hall, 2016), IBM relationship manager, is duidelijk geworden dat Watson is verwerkt in een aantal producten die aangeboden worden voor bedrijven. Dit houdt in dat deze producten opgesomd moeten worden om erachter te komen welk product het meest interessant is voor de implementatie van het gewenste prototype. Hierdoor is de eerste deelvraag aangepast:

"Wat is IBM Watson en welke producten worden aangeboden vanuit IBM Watson?"

Om een keuze te maken in de gebruikte producten en de services die daarbij horen moet gedefinieerd worden welke functionele eigenschappen het prototype nodig heeft. Dit feit heeft geresulteerd in de volgende deelvraag:

"Welke functionele eigenschappen heeft het prototype nodig?"

Watson implementatie

Om te achterhalen hoe Watson geïmplementeerd kan worden is in eerste instantie naar een aantal open-source applicaties gekeken. De onderstaande applicaties zijn open-source applicaties die gebruik maken van één of meerdere Watson services. Deze applicaties tonen een aantal mogelijkheden van Watson. De applicaties in deze paragraaf zijn gemaakt met behulp van de binnen Bluemix geleverde Watson services en zijn te vinden via de IBM Watson Gallery (IBM, 2016c).

- Questions on the Natural Language Classifier
Deze applicatie maakt het mogelijk om in natuurlijke taal vragen te stellen over de Natural Language Classifier service. De "Questions on the Natural Language Classifier" applicatie is ontwikkeld in Java en deze applicatie maakt gebruik van één Watson service, namelijk de Natural Language Classifier service.
Demo: <http://watson-on-classifier.mybluemix.net/>
- What's in theaters?
Deze applicatie maakt het mogelijk om, met behulp van natuurlijke taal, huidige of verwachte films te vinden aan de hand van een aantal criteria. Net zoals de vorige applicatie is ook de "What's in theaters" applicatie ontwikkeld in Java. Deze applicatie heeft toegang tot de database van themoviedb.org om filmgegevens op te halen. Verder gebruikt deze applicatie twee Watson services, de Natural Language Classifier service en de Dialog service. De werking van deze applicatie is gedocumenteerd waardoor het in combinatie met de open-source code een goed voorbeeld is voor een casestudy.
Demo: <http://watson-movieapp-nlcdialog.mybluemix.net/watson-movieappdialog/dist/index.html#/>
- News explorer
Deze applicatie maakt automatisch een nieuws informatienetwerk en toont grote hoeveelheden nieuwsresultaten. De "News explorer" applicatie is ontwikkeld in Java en maakt gebruik van de AlchemyAPI. De AlchemyAPI is een onderdeel van Watson wat een set aan services levert. Deze set is vooral gefocust op het ophalen van informatie uit, op internet beschikbare, nieuwsartikelen en blogs.
Demo: <http://news-explorer.mybluemix.net/>

De demoapplicaties maken, binnen het Bluemix product, gebruik van verschillende Watson services. Daarom moet voor het gekozen product onderzocht worden welke services beschikbaar zijn en welke van deze services relevant zijn voor het prototype. Dit resulteert in de deelvraag:

"Welke services zijn beschikbaar gesteld door IBM Watson, en welke van deze services zijn relevant voor dit project?"

Watson API's

Duidelijk is geworden dat Watson benaderd kan worden met behulp van een aantal verschillende services. Elke service heeft zijn eigen REST-API beschikbaar waarmee de desbetreffende service gebruikt kan worden.

Nadat duidelijk is gemaakt welke services van belang zijn voor het prototype kan onderzocht worden hoe deze services werken. Hiervoor is de volgende deelvraag toegevoegd:

"Hoe kunnen de relevante services van IBM Watson benaderd worden en welke configuratiemogelijkheden hebben deze services?"

In de demoapplicaties is te zien dat meerdere services gebruikt kunnen worden om een werkend geheel te realiseren. Wanneer meerdere services relevant zijn voor het prototype moet gekeken worden hoe deze services aan elkaar geknoopt kunnen worden om een werkend geheel te realiseren. Dit resulteert in de volgende deelvraag:

"Hoe kunnen de relevante services aan elkaar geknoopt worden om een vraag- en antwoordsysteem te realiseren?"

Het kan voorkomen dat functionaliteit geïmplementeerd moet worden wat niet aangeboden wordt door IBM. Hiervoor is de onderstaande deelvraag toegevoegd:

"Welke extra functionaliteit, niet ondersteund door IBM Watson, moet ontwikkeld worden en hoe wordt dit gedaan?"

Training

Duidelijk is dat Watson trainingsdata nodig heeft. Voor de volledigheid van het onderzoek wordt aan het eind opgesomd welke eisen gesteld worden aan de trainingsdata. Hiervoor is de volgende deelvraag opgesteld:

"In hoeverre dient IBM Watson getraind te worden om kennis op te doen over de verzekeringsbranche?"

3.4 Definitieve deelvragen

In paragraaf 3.2 zijn een aantal voorlopige deelvragen gegeven. In paragraaf 3.3 zijn deze voorlopige deelvragen aangepast tot concrete deelvragen.

Tijdens dit vooronderzoek zijn de onderstaande definitieve deelvragen opgesteld:

- 1. Wat is IBM Watson en welke producten worden aangeboden vanuit IBM Watson?**
- 2. Welke functionele eigenschappen heeft het prototype nodig?**
- 3. Welke services zijn beschikbaar gesteld door IBM Watson, en welke van deze services zijn relevant voor dit project?**
- 4. Hoe kunnen de relevante services van IBM Watson benaderd worden en welke configuratiemogelijkheden hebben deze services?**
- 5. Hoe kunnen de relevante services aan elkaar geknoopt worden om een vraag- en antwoordsysteem te realiseren?**
- 6. Welke extra functionaliteit, niet ondersteund door IBM Watson, moet ontwikkeld worden en hoe wordt dit gedaan?**
- 7. In hoeverre dient IBM Watson getraind te worden om kennis op te doen over de verzekeringsbranche?**

4. Opdrachtschrijving

In de inleiding en de voorgaande paragrafen is al aangegeven dat gewerkt wordt aan een Watson implementatie binnen de verzekeringsbranche. Dit hoofdstuk zal een gedetailleerdere beschrijving geven van de opdracht en een voorbeeld geven van het gewenste resultaat. Paragraaf 4.1 zal de opdracht beschrijven. De onderzoekfase zal beschreven worden in paragraaf 4.2 en tot slot wordt de ontwikkelfase beschreven in paragraaf 4.3.

4.1 Beschrijving

De opdracht bestaat uit het ontwikkelen van een prototype waarmee de uiteindelijke klanten van Colimbra, met behulp van een dialoog, verzekeringsvragen kunnen stellen. In de inleiding is al aangegeven dat deze opdracht bestaat uit een onderzoeksfase en een ontwikkelfase. De onderzoeksfase zal als doel hebben het verduidelijken van de werking van IBM Watson. Nadat deze fase is afgerond zal duidelijk zijn op welke wijze Watson ingezet kan worden en zal aan de hand van dit onderzoek een prototype worden ontwikkeld.

4.2 Onderzoeksfase

Tijdens de onderzoeksfase wordt de werking van Watson onderzocht en wordt gekeken op welke manier Watson verzekeringsvragen kan beantwoorden.

De volgende hoofdvraag staat centraal binnen dit onderzoek:

Op welke manier kunnen vragen binnen de verzekeringsbranche, in natuurlijke taal, beantwoord worden met behulp van IBM Watson en hoe kan de werking hiervan bewezen worden met een prototype (proof of concept)?

Om deze hoofdvraag te beantwoorden is deze opgedeeld in deelvragen. De manier waarop deze vragen tot stand zijn gekomen is beschreven paragraaf 3.3.

1. Wat is IBM Watson en welke producten worden aangeboden vanuit IBM Watson?

Om de hoofdvraag te kunnen beantwoorden zal eerst beschreven worden wat Watson inhoudt en wat het doel is van Watson. Een antwoord op deze vraag wordt geformuleerd door te kijken naar beschikbare informatie op de IBM Watson website. Ook is informatie beschikbaar via conference calls met IBM.

2. Welke functionele eigenschappen heeft het prototype nodig?

Om te achterhalen welke Watson services nodig zijn zal eerst gedefinieerd worden welke functionele eigenschappen gewenst zijn voor het prototype. Het antwoord op deze deelvraag is tot stand gekomen door gesprekken met de opdrachtgever.

3. Welke services zijn beschikbaar gesteld door IBM Watson, en welke van deze services zijn relevant voor dit project?

Een korte beschrijving van elke service is nodig om te achterhalen welke Watson services daadwerkelijk nodig zijn. Deze vraag zal beantwoord worden met behulp van de beschikbare servicedocumentatie. Om te achterhalen welke van deze services nodig zijn wordt de beschrijving vergeleken met de functionele eigenschappen uit de vorige deelvraag.

4. Hoe kunnen de relevante services van IBM Watson benaderd worden en welke configuratiemogelijkheden hebben deze services?

Wanneer de vorige vragen zijn beantwoord is duidelijk welke services uiteindelijk gebruikt moeten worden voor de ontwikkeling van een prototype. Voor deze services wordt beschreven hoe ze werken, hoe ze benaderd kunnen worden en in hoeverre elke service geconfigureerd kan worden voor een specifiek scenario. Ook deze vraag wordt beantwoord met behulp van de servicedocumentatie.

5. Hoe kunnen de relevante services aan elkaar geknoopt worden om een vraag- en antwoordsysteem te realiseren?

Om het vraag- en antwoordsysteem te realiseren moeten de verschillende services met elkaar samenwerken. Daarom zal een ontwerp gemaakt moeten worden waarin wordt beschreven welke gegevens van welke services beschikbaar worden gesteld voor een andere service en/of gebruiker. Het beantwoorden van deze vraag zal resulteren in een schema waarin de architectuur van het vraag- en antwoordsysteem getoond wordt. Aangezien de documentatie niet ingaat op het combineren van meerdere services wordt deze vraag beantwoord aan de hand van een bestudeerd voorbeeldapplicatie, en informatie verschaft uit conference calls met IBM.

6. Welke extra functionaliteit, niet ondersteund door IBM Watson, moet ontwikkeld worden en hoe wordt dit gedaan?

Het is mogelijk dat niet alle gewenste functionaliteit geleverd wordt met de services van Watson. Daarom zal beschreven worden welke functionaliteit mist en dus zelf gerealiseerd moet worden. Om deze vraag te beantwoorden worden de functionele eigenschappen vergeleken met de architectuur van het vraag- en antwoordsysteem.

7. In hoeverre dient IBM Watson getraind te worden om kennis op te doen over de verzekeringsbranche?

Watson dient getraind te worden binnen een specifieke sector voordat het vragen kan beantwoorden over die sector. In dit geval zal Watson getraind moeten worden om de verzekeringsbranche te begrijpen. Het doel van deze deelvraag is om te beschrijven hoeveel trainingsdata beschikbaar moet zijn voor de ontwikkeling van het prototype en welke inhoud deze trainingsdata moet bevatten. Deze vraag zal beantwoord worden met behulp van de beschikbare servicedocumentatie.

4.3 Ontwikkelfase

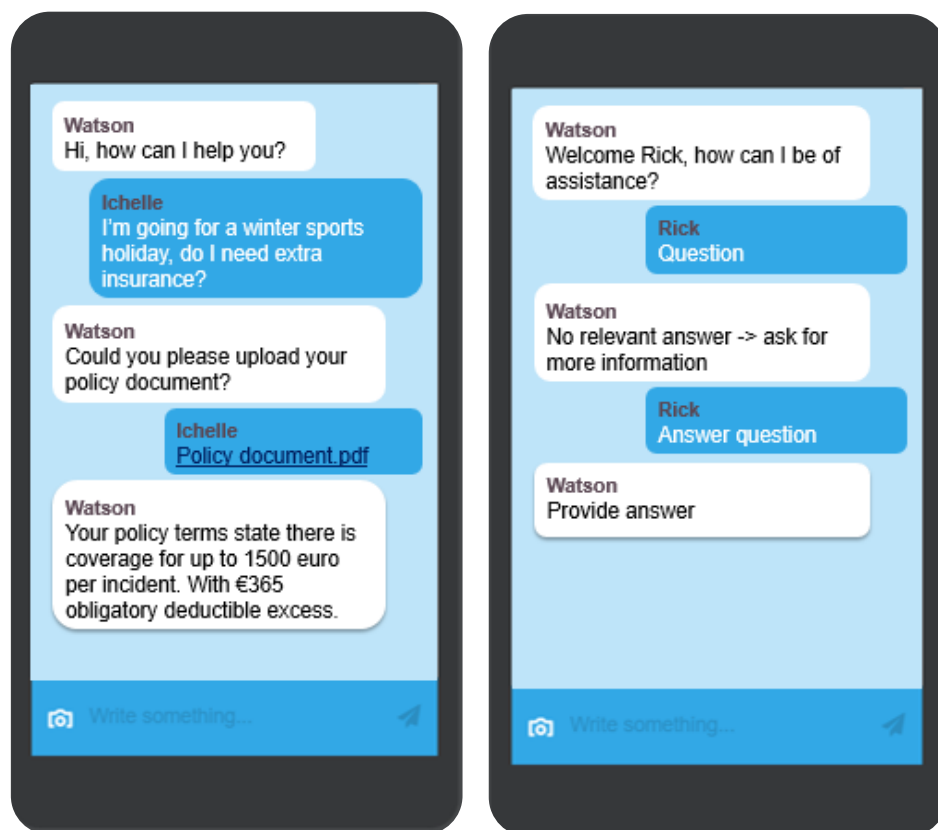
Het prototype dat gerealiseerd wordt zal inhoudelijk afhangen van het onderzoek. De focus voor dit prototype ligt op het bewijzen van de werking van Watson. De wens is hier dat het prototype in ieder geval een aantal vragen over een autoverzekering kan beantwoorden. Dit prototype zal eind juni worden gepresenteerd op een verzekeringsbeurs. De specifieke functionele eigenschappen van dit prototype wordt zijn gedefinieerd in de onderzoeksfase (paragraaf 6.2).

Aangezien het doel van het prototype is om de werking van Watson te toetsen is dit waar de focus op ligt. Echter is, aangezien dit prototype gepresenteerd wordt, een persenteerbare interface gewenst.

Figuur 1 is toegevoegd om een beeld te geven over de gewenste functionaliteit. Deze figuur geeft niet aan hoe de interface er uiteindelijk uit komt te zien. Het prototype zal wel een soortgelijk chatinterface moeten hebben. De manier waarop deze interface wordt geïmplementeerd staat vrij.

De rechterhelft laat een mogelijke conversatie zien waarin een vraag wordt gesteld over de dekking tijdens wintersport. In dit geval heeft Watson nog geen polis document van de gebruiker dus vraagt Watson of de gebruiker haar polis document kan uploaden. Dit rechter voorbeeld is overigens overgenomen uit de business case, opgesteld door (Kleeff, 2015).

De linkerhelft geeft een conversatieverloop weer. In dit geval heeft Watson niet genoeg informatie dus wordt een vraag teruggesteld waarna een antwoord gegeven kan worden op de initiële vraag.



Figuur 1 Implementatie voorbeeld

5. IBM Watson

In dit hoofdstuk wordt een beschrijving gegeven van Watson. Watson is een supercomputer van IBM die de natuurlijke taal begrijpt. Dit hoofdstuk zal dieper ingaan op het doel van Watson en de mogelijkheden van Watson. Een groot deel van deze informatie is eerder verzameld voor het plan van aanpak, toegevoegd als bijlage A. De informatie uit dat plan van aanpak is afkomstig uit de officiële IBM-website (IBM, 2016b). Ook is een deel van deze informatie al gegeven in het theoretisch kader, in hoofdstuk 3.

Dit hoofdstuk geeft een antwoord op deelvraag 1:

Wat is IBM Watson en welke producten worden aangeboden vanuit IBM Watson?

In paragraaf 5.1 zal begonnen worden met een korte beschrijving van het bedrijf International Business Machines (IBM). Paragraaf 5.2 zal verder gaan met een beknopte beschrijving van Watson. Hierin wordt aangegeven hoe Watson is ontstaan. In paragraaf 5.3 wordt een opsomming gemaakt van de verschillende Watson producten. Tot slot zal in paragraaf 5.4 worden aangegeven wat de invloed van de conclusie van deze paragraaf is, op het vervolg van dit onderzoek.

5.1 International Business Machines (IBM)

International Business Machines (IBM) is een Amerikaans bedrijf dat zich bezighoudt met de ontwikkeling van hardware, middleware en software, en met het bieden van infrastructuur, hosting en consultancy services.

IBM biedt bedrijven de mogelijkheid om partner te worden van het Watson ecosysteem. Hierdoor wordt een "shared revenues partnership" opgezet tussen beide partijen. Dit houdt in dat beide bedrijven investeren in de ontwikkeling van het product en uiteindelijk de winst zullen delen. Colimbra is dit partnerschap aangegaan waardoor Colimbra onder andere toegang heeft tot de producten aangeboden vanuit het Watson ecosysteem.

5.2 Watson

Watson is een supercomputer van IBM die de natuurlijke taal begrijpt en getraind kan worden om een bepaalde sector, in dit geval de verzekeringsbranche, te begrijpen. Dit gebeurt door de mogelijkheid van Watson om documenten of webpagina's om te zetten naar kennis. Uiteindelijk kan Watson getraind worden als "expert" binnen de verzekeringsbranche, wat het onder andere mogelijk maakt om Watson in te zetten als een digitale helpdesk.

Het bedrijf IBM heeft een aantal momenten waarop een "Grand Challenge" wordt aangegaan. Tijdens deze uitdagingen wordt ontwikkeld aan machines die tegenover de mens worden gezet. Eén van deze uitdagingen heeft geresulteerd in de ontwikkeling van Deep Blue. Deep Blue is de machine die de wereldkampioen schaken, Garry Kasparov, heeft verslagen in 1997 (Campbell, 2002).

Watson is net zoals Deep Blue ontwikkeld als "Grand Challenge". Deze keer is Watson gericht op het winnen van "Jeopardy!", een Amerikaans tv-programma waarin een quiz wordt gespeeld. Tijdens deze quiz worden antwoorden gegeven waarna de deelnemers de bijbehorende vraag moeten verzinnen. Om deze quiz te winnen moest Watson dus eerst het antwoord begrijpen en daar daarna de bijbehorende vraag bij formuleren. Watson heeft uiteindelijk gewonnen van de, in die tijd, twee beste spelers die hebben meegespeeld: Ken Jennings en Brad Rutter (Best, 2015).

5.3 Producten

Het in paragraaf 5.1 genoemde Watson ecosysteem biedt toegang tot een aantal Watson producten, namelijk de volgende:

- Watson Engagement Advisor;
- Watson Explorer;
- Watson Discovery Advisor;
- Watson for Oncology;
- Watson for Clinical Trial Matching;

Buiten deze producten, die alleen toegankelijk zijn voor IBM-partners, is een verzameling van soortgelijke services via IBM Bluemix.

Watson Engagement Advisor

Het product dat voor Colimbra het meest interessant is, is de Watson Engagement Advisor. De Watson Engagement Advisor is een technologie waarmee gecommuniceerd wordt met de gebruikers, luisterend naar vragen om een oplossing aan te bieden. Het is een vraag- en antwoordsysteem, de gebruiker kan vragen stellen in natuurlijke taal. Watson begrijpt wat de gebruiker wil doen, analyseert de context en presenteert daarna de meest relevante oplossing. Dit product is het meest geschikt voor dit project aangezien het inzetten van Watson als digitale helpdesk als doel heeft om vragen te beantwoorden.

Watson Explorer

Het Watson Explorer product implementeert een platform waarmee gestructureerde en ongestructureerde data/documenten opgezocht en geanalyseerd kunnen worden. Dit maakt het mogelijk om een overzicht op te stellen van verschillende databronnen.

Watson Discovery Advisor

De Watson Discovery Advisor kan ingezet worden binnen een domein om patronen te herkennen en nieuwe patronen aan te bieden. Denk hierbij aan het creëren van nieuwe recepten of het zoeken naar de juiste medische behandeling.

Watson for Oncology & Watson for Clinical Trial Matching

De Watson for Oncology en de Watson for Clinical Trial Matching producten zijn beide specifiek ontwikkeld voor het medische domein. Hierdoor worden deze producten niet gebruikt voor de implementatie van het gewenste prototype.

5.4 Prototype

In paragraaf 5.3 is aangegeven dat het Watson Engagement Advisor product interessant is voor de realisatie voor dit prototype. Uit een conference call met (Ravencroft, 2016) is duidelijk geworden dat dit product echter niet meer wordt aangeboden, de services van Bluemix zijn en worden verbeterd om als vervanging te kunnen dienen.

Hierdoor is Colimbra in een soort bèta programma terechtgekomen, waarin verschillende aspecten nog onduidelijk zijn en bepaalde services nog niet volledig gedocumenteerd zijn. Onduidelijk is bijvoorbeeld welke Bluemix services nodig zijn en hoe deze aan elkaar gekoppeld kunnen worden om een vraag- en antwoordsysteem, zoals de hiervoor genoemde Engagement Advisor, te kunnen realiseren.

Door deze conclusie zal gedurende het verloop van dit onderzoek niet met de aangeboden producten vanuit het Watson ecosysteem gewerkt worden maar met de services aangeboden via IBM Bluemix.

6. Watson use case

Hiervoor is meerdere keren aangegeven dat het systeem instaat moet zijn om in natuurlijke taal gestelde vragen te beantwoorden met een in natuurlijke taal geformuleerd antwoord. Maar op welke manier moet dit gebeuren en welke onderwerpen moet het systeem begrijpen? Deze functionele eigenschappen worden in dit hoofdstuk vastgelegd. De modellen in dit hoofdstuk zijn ook vastgelegd in het functioneel ontwerp, toegevoegd als bijlage B.

Dit hoofdstuk geeft een antwoord op deelvraag 2:

Welke functionele eigenschappen heeft het prototype nodig?

Paragraaf 6.1 zal een beschrijving geven van de gewenste applicatie. Het volledig implementeren van de applicatie zoals beschreven in paragraaf 6.1 is niet haalbaar. Gedurende de afstudeerperiode zal een eerste prototype worden ontwikkeld, de functionele eigenschappen van dit prototype worden beschreven in paragraaf 6.2.

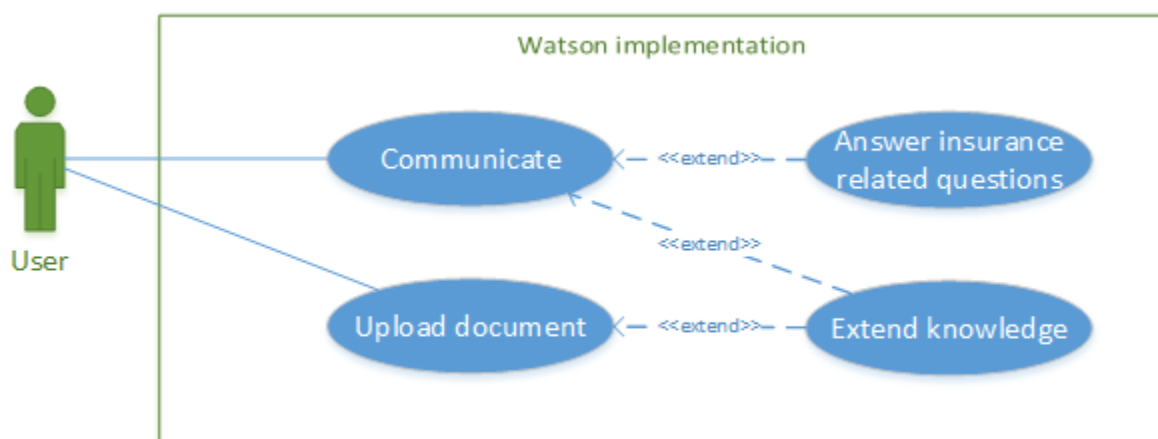
6.1 Beschrijving

Colimbra wil een applicatie realiseren waarmee gebruikers in staat zijn om met Watson te communiceren. Uiteindelijk moet Watson, door zijn kennis over de verzekeringsbranche, in staat zijn om een helpdeskmedewerker te vervangen.

Van Watson wordt verwacht dat het over basiskennis beschikt. Wanneer bijvoorbeeld naar het weer wordt gevraagd begrijpt Watson dit en hoort Watson te reageren met het feit dat het gespecialiseerd is in verzekeringen en niet beschikt over weersinformatie (tenzij Watson het weer kan opzoeken).

Een verzekeraar kan verschillende verzekeringen aanbieden zoals een autoverzekering en een levensverzekering. Watson moet in staat zijn om te begrijpen over welke verzekering een gebruiker zijn vragen stelt.

Het mooie aan de implementatie van Watson is dat het gebruik van de applicatie eenvoudig is. De gebruiker praat simpelweg met het systeem en in sommige gevallen kan een document opgestuurd worden. Dit resulteert in het simpele use case diagram in figuur 2.



Figuur 2 Use case diagram

6.2 Prototype

In de vorige paragraaf is beschreven hoe Colimbra uiteindelijk de Watson implementatie wil realiseren. De uitvoering van dit project is een eerste stap om de gewenste applicatie te realiseren.

Het volledige systeem kan niet binnen de afstudeerperiode gerealiseerd worden. In eerste instantie omdat er niet genoeg data beschikbaar is om de applicatie te realiseren zonder een verzekeraar te benaderen. Bovendien is de afstudeerder geen domeinexpert maar een ontwikkelaar, daarom zal de afstudeerder zich vooral focussen op het opzetten van het vraag- en antwoordsysteem en niet zo zeer op de hoeveelheid vragen die correct beantwoord kunnen worden.

Voor het eerste prototype wordt uitgegaan van één verzekeraar die alleen een autoverzekering aanbiedt. Verder zijn geen polisdocumenten beschikbaar dus deze worden buiten beschouwing gelaten.

Figuur 3 is afkomstig uit het plan van aanpak (bijlage A) en bevat een uitwerking van de MoSCoW methode. Hierin worden de Must, de Should, de Could en de Won't have's van het prototype aangegeven. Deze MoSCoW methode is aan de hand van het volgende hoofdstuk aangevuld met de bijbehorende Watson service(s).

MoSCoW		Watson service(s)
Must	Dialogoog om de gebruiker te begeleiden	Dialog
	Mogelijkheid om polisdocumenten en polisvoorwaarden te uploaden	Document Conversion
	Informatie uit beschikbaar gestelde documenten moet te benaderen zijn via de dialogoog.	Retrieve & Rank
Should	Beter begrip van wat de gebruiker wil bereiken (de dialogoog service let alleen op zinsopbouw met behulp van keywords. Met deze toevoeging zal de intentie van de input worden begrepen waardoor beter gereageerd kan worden).	Natural Language Classifier
	Chatinterface om Watson te benaderen.	-
Could	Gesproken aansturing	Speech to Text en Text to Speech
	Admin interface om Watson te trainen	-
	Het bijhouden van gebruik in een database om Watson op een later stadium verder te kunnen trainen.	-
Won't	Ondersteuning voor de Nederlandse taal (op dit moment niet ondersteund door Watson).	-
	Socialmedia analyse om informatie over de gebruiker te verzamelen	AlchemyAPI en/of Personality Insights
	Object herkenning om geüploade plaatjes te begrijpen	Visual recognition

Figuur 3 MoSCoW methode

7. Watson serviceoverzicht

Dit hoofdstuk zal alle 18 beschikbare Watson services kort beschrijven. Bij elke beschrijving wordt aangegeven of de desbetreffende service van belang is voor het prototype. Indien dit niet het geval is wordt beredeneerd of de service een mogelijke toevoeging kan bieden voor in de toekomst. Alle informatie uit deze paragraaf is afkomstig uit de officiële IBM Watson documentatie (IBM, 2016d).

Dit hoofdstuk geeft uiteindelijk een antwoord op deelvraag 3:

Welke services zijn beschikbaar gesteld door IBM Watson, en welke van deze services zijn relevant voor dit project?

Door de grote omvang van het serviceoverzicht is deze toegevoegd als bijlage D. De beschikbare services worden genoemd in paragraaf 7.1 en in paragraaf 7.2 wordt aangegeven welke van deze services relevant zijn voor het eerste prototype.

7.1 Beschikbare services

Van de 18 beschikbare services worden in de bijlage 15 beschreven. De overige services zijn "third-party" bèta services welke onvoldoende zijn gedocumenteerd. Buiten de documentatie zijn de functionele eigenschappen van deze services niet van belang voor het prototype.

De volgende 15 services zijn beschreven:

- AlchemyAPI;
- Concept Expansion (open bèta);
- Concept Insights;
- Dialog;
- Document Conversion;
- Language Translation;
- Natural Language Classifier;
- Personality Insights;
- Relationship Extraction (open bèta);
- Retrieve and Rank;
- Speech to Text & Text to Speech;
- Tone Analyzer;
- Tradeoff Analytics;
- Visual Recognition (open bèta).

Elke service hierboven genoemd beschikt over een demo. Een link naar de demo van elke service is toegevoegd aan het serviceoverzicht in bijlage D. Per service is gekeken naar waar de service voor dient en wat met de desbetreffende service gerealiseerd kan worden.

Hieronder wordt het gebruik voor van de services voor Colimbra herhaald. Een beschrijving van de werking is, zoals eerder vermeld, te vinden in bijlage D.

AlchemyAPI

De AlchemyAPI service kan een grote hoeveelheid informatie verzamelen uit tekst en plaatjes. Het verzamelen van deze informatie is belangrijk voor het prototype maar hiervoor wordt geen gebruik gemaakt van deze services. Dit omdat de AlchemyAPI services vooral zijn gericht op het analyseren van tekst afkomstig uit nieuws gerelateerde webpagina's. Voor het extraheren van de benodigde informatie uit tekst worden voor het prototype andere services gebruikt (Natural Language Classifier & Retrieve and Rank).

Concept Expansion

Deze service zal niet gebruikt worden voor het prototype. Het zou in de toekomst wel gebruik kunnen worden om een verzekeringsgericht woordenboek op te zetten waardoor andere services een beter resultaat kunnen leveren.

Concept Insights

Deze service wordt niet gebruikt in het prototype. In de toekomst kan deze service gebruikt worden om bijvoorbeeld extra achtergrondinformatie te vinden over verschillende verzekeringen of om een antwoord op een vraag te zoeken via Wikipedia.

Dialog

Aangezien door deze service een applicatie in staat is om in natuurlijke taal te communiceren met de gebruiker wordt deze service gebruikt in het prototype. Door deze service kan het antwoord op regelmatig gestelde vragen vooraf gedefinieerd worden.

Document Conversion

Voor Colimbra is documentconversie niet alleen mogelijk via deze service aangezien Colimbra ook een eigen oplossing heeft voor documentconversie. Voor het prototype zal echter wel gebruik gemaakt worden van deze service omdat deze service het converteren naar "answer units" ondersteund.

Language Translation

Deze service zal alleen gebruikt worden door Colimbra wanneer de Nederlandse taal wordt ondersteund. Wanneer de Nederlandse taal ondersteund wordt kunnen Engelstalige bronnen gebruikt worden om een Nederlandstalig antwoord te geven en andersom.

Natural Language Classifier

Het is de bedoeling dat het prototype volledig wordt bestuurd in natuurlijke taal. Deze service wordt gebruikt om het resultaat van de applicatie te verbeteren door de intentie van een stuk tekst te kunnen achterhalen. Verschillende classificatieinstanties kunnen gebruikt worden voor bijvoorbeeld het classificeren of een vraag wel of niet een verzekeringsvraag is en een andere om te achterhalen wat voor soort antwoord de gebruiker verwacht, vaak ook de "lexical answer type" genoemd.

Personality Insights

Voor het prototype zal deze service niet worden gebruikt. In de toekomst kan Colimbra deze service inzetten om bijvoorbeeld, bij klanten die daar toestemming voor geven, de social media accounts te doorlopen om met de verzamelde gegevens een persoonlijkheidsportret op te stellen. Dit kan dan weer gebruikt worden om meer persoonlijke interacties met gebruikers te realiseren.

Relationship Extraction

De Relationship Extraction service wordt niet gebruikt in het prototype. Een mogelijk gebruiksscenario is om nieuws afkomstig uit de verzekeringsbranche te verzamelen.

Retrieve and Rank

Deze service zal gebruikt worden voor het prototype. Deze service zorgt ervoor dat de applicatie door bijvoorbeeld de polisdocumenten kan zoeken voor een antwoord. Ook kan deze service ingezet worden om bijvoorbeeld relevante informatie uit Wikipedia te tonen.

Speech to Tekst & Tekst to Speech

Deze services zullen in eerste instantie niet gebruikt worden in het prototype. Aangezien de implementatie van deze service waarschijnlijk vrij eenvoudig is kan dit wel toegevoegd worden aan het prototype. Hier ligt echter niet de focus op.

Tone Analyzer

Deze service wordt niet gebruikt in het prototype. In de toekomst kan deze service gebruikt worden om bijvoorbeeld te achterhalen waar de meest frustrerende vragen over gaan. Of om, indien nodig, de gebruiker met behulp van een dialoog op zijn gemak te kunnen stellen.

Tradeoff Analytics

Deze service wordt niet gebruikt in het prototype. In de toekomst kan deze service gebruikt worden om gebruikers te helpen met het kiezen van een verzekering.

Visual Recognition

Deze service wordt niet gebruikt in het prototype. In de toekomst kan deze service gebruikt worden om, door de uiteindelijke klant, geüploade documenten te begrijpen en hier een passende reactie op te geven.

7.2 Relevante services

In de voorgaande paragrafen is aangegeven dat de onderstaande services gebruikt zullen worden tijdens de realisatie van het prototype:

- Dialog
- Document conversion
- Natural language classifier
- Retrieve and rank

De Dialog service zorgt voor een begeleiding van de conversatie. Met deze service kunnen vragen gesteld worden aan de gebruiker wanneer meer informatie nodig is.

De Document Conversion service zorgt ervoor dat documenten omgezet worden naar een formaat waarmee de Retrieve and Rank service kan werken.

De Natural Language Classifier service zorgt ervoor dat de intentie van de gebruiker wordt achterhaald.

De Retrieve and Rank service zorgt ervoor dat informatie uit bijvoorbeeld een document kan worden geciteerd.

8. Servicebeschrijving

Dit hoofdstuk zal de, voor het prototype, relevante services in detail beschrijven. De API van deze services wordt bestudeerd en de benodigde (trainings-)data wordt beschreven. Informatie uit dit hoofdstuk is afkomstig uit de officiële IBM Watson documentatie (IBM, 2016d).

Dit hoofdstuk geeft een antwoord op deelvraag 4:

Hoe kunnen de relevante services van IBM Watson benaderd worden en welke configuratie mogelijkheden hebben deze services?

Relevante services

In paragraaf 7.2 is aangegeven welke services relevant zijn voor het prototype. Deze services zijn hieronder opgesomd met een korte beschrijving en de reden van relevantie zoals aangegeven in de vorige paragraaf.

- **Dialog**
De Dialog service zorgt ervoor dat applicaties begeleid kunnen worden aan de hand van een vooraf gedefinieerd dialoogscript. Zoals in hoofdstuk 6.1 is beschreven moet de applicatie met de gebruiker kunnen communiceren in natuurlijke taal. Deze service helpt met het beantwoorden van vaak voorkomende vragen. Gebruiksgegevens over deze service wordt gegeven in paragraaf 8.1.
- **Document conversion**
De applicatie moet in staat zijn om verzekeringsdocumenten te begrijpen. Om deze documenten te kunnen verwerken moeten ze eerst omgezet worden in "answer units" welke gebruikt kunnen worden door de Retrieve and Rank service. Informatie over de Document Conversion service is gegeven in paragraaf 8.2.
- **Natural Language Classifier**
Classificatie van tekst is belangrijk om te kunnen achterhalen wat de uiteindelijke gebruiker verwacht en om te achterhalen waar een stuk tekst over gaat. Gebruiksgegevens en de benodigde trainingsdata wordt gegeven in paragraaf 8.3.
- **Retrieve and Rank**
De Retrieve and Rank service zorgt ervoor dat de, naar "answer units" geconverteerde, verzekeringsdocumenten doorzocht kunnen worden. Hierdoor kan een stuk tekst uit de vooraf verwerkte documenten worden geciteerd. Gebruiksgegevens en de benodigde trainingsdata wordt beschreven in paragraaf 8.4.

Configuratiemogelijkheden

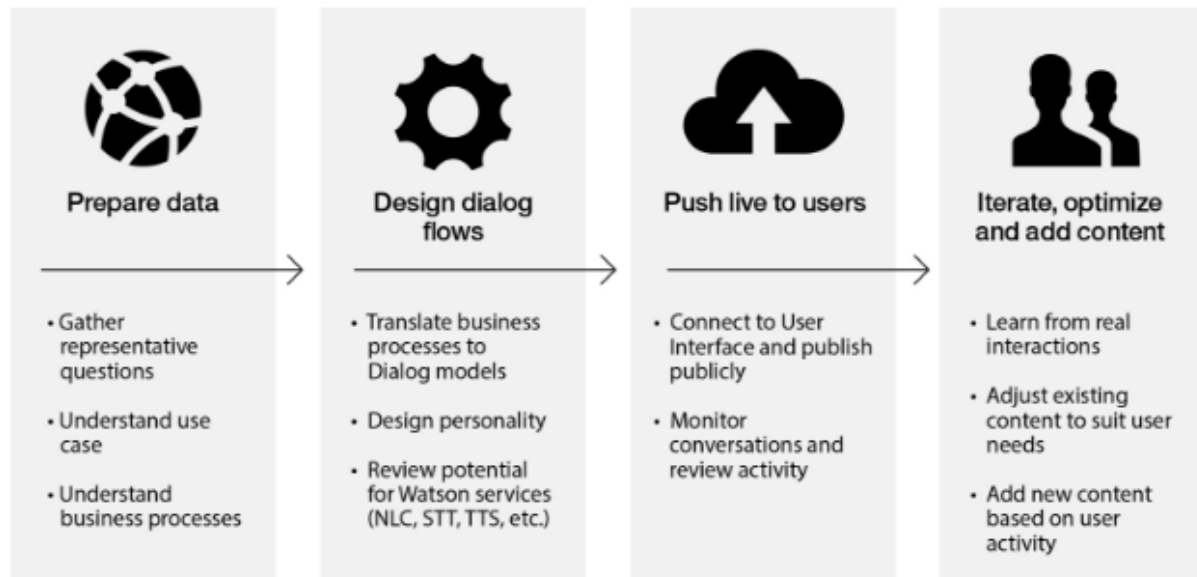
Elke Watson service is te benaderen met behulp van een REST API. In paragraaf 8.5 wordt een samenvatting gegeven over de configuratiemogelijkheden van elke service. Het opzetten van een vraag- en antwoordsysteem met behulp van deze services is beschreven in hoofdstuk 9.

8.1 Dialog

De Watson Dialog service is bedoeld om een gebruiker in natuurlijke taal te kunnen begeleiden met behulp van een dialoog.

Workflow

Figuur 4 geeft weer welke workflow gebruikelijk is bij het implementeren en inzetten van de Dialog service. In deze workflow zijn stap drie en vier iteratief.



Figuur 4 Dialog service - implementatie workflow (IBM, 2016d)

Allereerst moet besloten worden waar de Dialog service voor wordt ingezet. Welke vragen moeten beantwoord kunnen worden? Welke bedrijfsprocessen moeten verwerkt worden?

De tweede stap is om deze data te vertalen naar een dialoogmodel. Dit model geeft aan op welke manier het systeem moet reageren op een bepaalde input. In dit script kan ook een bepaalde persoonlijkheid voor het systeem worden gecreëerd.

Tijdens stap drie wordt de applicatie gepubliceerd. Het dialoogsript wordt geüpload naar de Dialog service en de service wordt gekoppeld aan een interface waarin gebruikers met het systeem kunnen communiceren. Het systeem is nu benaderbaar door gebruikers. De conversaties die gebruikers hebben met het systeem worden gelogd en beoordeeld.

In stap vier wordt de applicatie geoptimaliseerd en uitgebreid. De gelogde gegevens worden gebruikt om het systeem uit te breiden en aan te passen. Nieuwe functionaliteit wordt toegevoegd aan de hand van de gebruiksinformatie.

Dataverzameling

De Dialog service zal het gesprek laten verlopen met behulp van een vooraf gedefinieerd script. Dit houdt in dat voorafgaand aan de ontwikkeling eerst bedacht moet worden welke mogelijke conversaties een gebruiker met het systeem kan hebben. Een mogelijk dialoogscript zou verdeeld kunnen worden over de volgende categorieën: "verzekeringsvragen", "acties" en "overig".

De verzekeringsvragen zijn domein specifiek en het meest uitgebreid. Hierin zullen de meest voorkomende verzekeringsvragen vooraf worden gedefinieerd. Ook worden hier vragen opgenomen waar geen direct antwoord op te geven is, voor het beantwoorden van deze vragen is meer context nodig waardoor het systeem wedervragen moet stellen om meer context te verzamelen. Vragen die niet in dit script zijn gedefinieerd worden met behulp van de andere Watson services beantwoord.

Acties zijn per verzekeringsmaatschappij verschillend. De verzekeringsmaatschappij waar deze service wordt ingezet kan een aantal acties laten verlopen via Watson. Dit zijn acties zoals bijvoorbeeld het veranderen van een wachtwoord, het aanpassen van de woonlocatie, of het instellen van een bijnaam. Het dialoogscript kan ingericht worden om de benodigde informatie te verzamelen en de actie uit te voeren.

De overige toegevoegde conversaties zijn niet domein specifiek en zijn toegevoegd om het systeem persoonlijker en interactiever te maken. Hierin staan bijvoorbeeld de reacties gedefinieerd voor wanneer het systeem wordt gegroet of wanneer naar het weer gevraagd wordt.

Dialoog ontwerp

De verzamelde gegevens moeten verwerkt worden in de dialoog. Hierin wordt ontworpen op welke manier het systeem met de gebruiker communiceert. Het dialoogscript wordt in XML-formaat gerealiseerd.

Een dialoogscript bestaat altijd uit een `dialog` tag. In deze `dialog` tag kunnen de volgende onderdelen eenmalig voorkomen: `flow`, `entities`, `constants`, `variables` en `settings`. De `flow` tag beschrijft de flow van de conversatie, binnen deze tag wordt gedefinieerd hoe de conversatie verloopt. De `entities` tag bevat een verzameling van woorden die onder een gespecificeerde context vallen. De `constants` en `variables` tags definiëren zoals de naam al aangeeft constanten en variabelen. De `settings` tag is bedoeld om de Dialog service in te stellen.

Publicatie

Het dialoogscript wordt met behulp van de REST API verstuurd naar de Dialog service. Wanneer dit succesvol is ingelezen kan gecommuniceerd worden met behulp van de REST API. Er is geen Watson service beschikbaar waarmee een interface wordt gemaakt. Aangezien een interface toch gewenst is voor het prototype moet hier een andere oplossing voor gevonden worden.

Optimalisatie en uitbreiding

De optimalisatie en uitbreidingsfase is afhankelijk van de verzamelde data. De verzamelde data wordt nagekeken om te beslissen welke niet gedefinieerde conversaties wel of niet opgenomen moeten worden in een nieuwe versie van het dialoogscript.

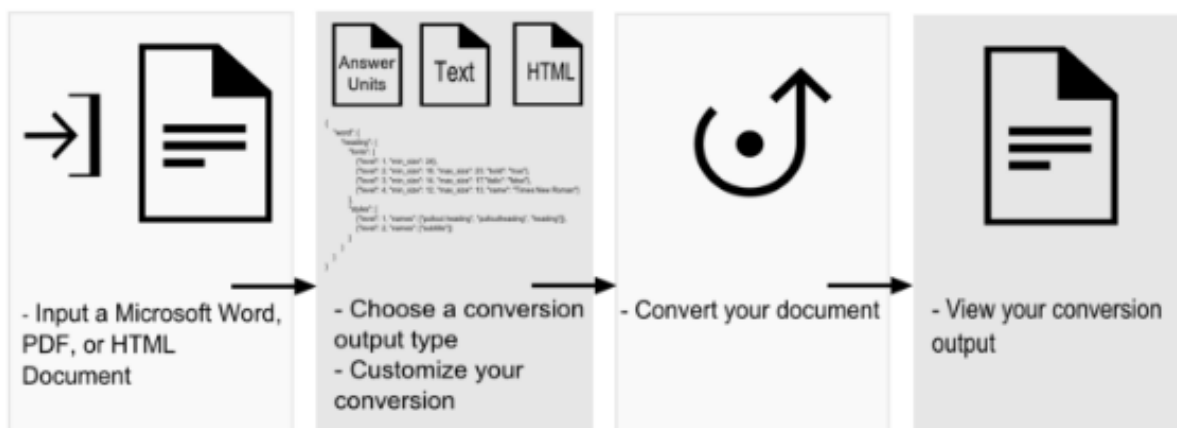
8.2 Document conversion

Zoals eerder aangegeven is de Document Conversion service een simpele conversie service zonder cognitieve vaardigheden. Deze paragraaf zal aangeven op welke manier de Document Conversion service ingezet kan worden.

De Document Conversion service converteert een HTML, PDF, of Word document. Dit document wordt omgezet naar een genormaliseerde HTML-pagina, een genormaliseerd tekstbestand of naar een JSON-geformatteerde answer units document.

Workflow

In figuur 5 wordt aangegeven welke workflow wordt gehanteerd bij het implementeren en inzetten van de Document Conversion service.



Figuur 5 Document conversion service - implementatie workflow (IBM, 2016d)

In het bovenstaande schema zijn vier stappen beschreven. Stap één is het kiezen van het document om te converteren. Het input document dient in Microsoft Word, PDF of HTML-formaat te staan. In stap drie wordt de conversie gestart en in stap vier wordt het geconverteerde document binnengehaald.

De meest interessante stap in dit schema is stap twee waarin wordt aangegeven op welke manier het document geconverteerd moet worden. Dit gebeurt aan de hand van een meegestuurde file in JSON-formaat.

Document conversion configuratie

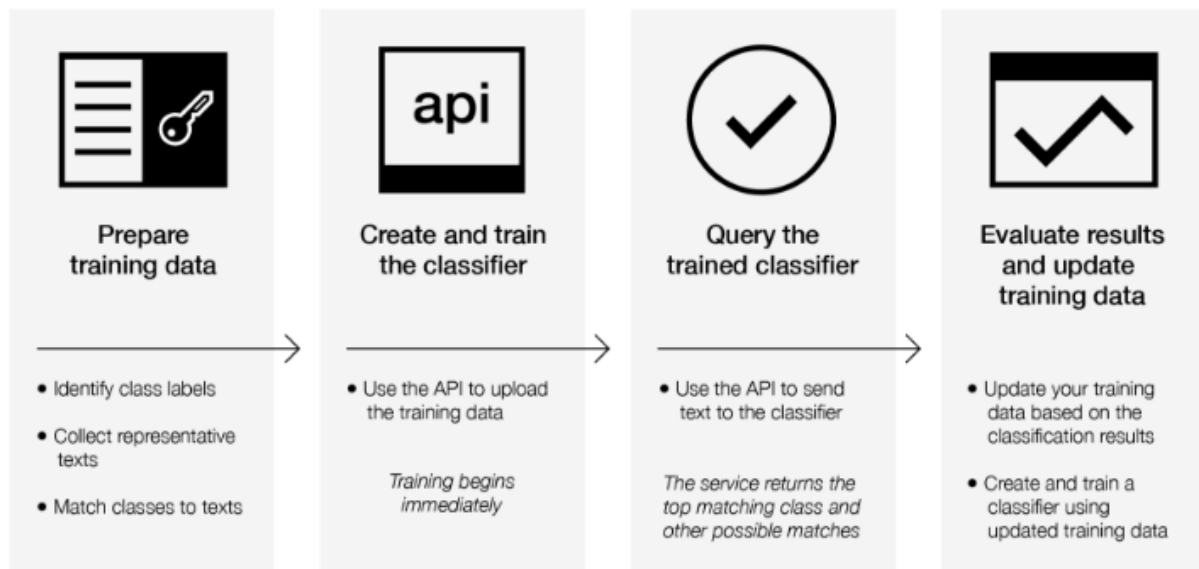
Het invoerdocument zal op basis van fontstijl worden gesegmenteerd in kopjes. Dit kan geconfigureerd worden met behulp van een JSON-document. De enige waarde die verplicht is in dit document is de `conversion_target`, deze waarde geeft aan naar welk formaat geconverteerd moet worden. Verder kan aangegeven worden welke kopjes meegenomen worden en per `conversion_target` kan aangegeven worden welke fonts bij welk kopje horen.

8.3 Natural Language Classifier

De Natural Language Classifier service kan ingezet worden om een stuk tekst te classificeren. De manier waarop dit gebeurt is afhankelijk van de trainingsdata.

Workflow

De onderstaande figuur (figuur 6) geeft aan op welke manier de Natural Language Classifier service aangemaakt en gebruikt kan worden. Dit schema gaat uit van één classifier, het is mogelijk om meerdere classifiers aan te maken door hetzelfde proces te doorlopen. De derde en vierde stap in deze workflow zijn iteratief.



Figuur 6 Natural language classifier service - implementatie workflow (IBM, 2016d)

De eerste stap is het aanmaken van de trainingsdata. Voor elke classifier moet besloten worden welke labels (classificaties) nodig zijn. De trainingsdata bestaat uit een collectie van tekst inclusief referentie naar het bijbehorende label. Deze collectie wordt gebruikt als trainingsdata.

Het trainen van de classifier gebeurt intern binnen IBM, na het opsturen van de trainingsdata wordt de training automatisch gestart. Voordat de classifier gebruikt kan worden moet de training voltooid zijn, de status hiervan kan opgevraagd worden.

Tijdens de derde stap wordt de classifier in gebruik genomen. Net zoals bij de Dialog service zal het gebruik van deze service gelogd worden om op een later stadium te beoordelen.

In stap vier wordt de verzamelde data gebruikt om de, in stap twee gecreëerde trainingsdata, aan te vullen. Met deze aangevulde trainingsdata wordt de classifier opnieuw aangemaakt waardoor hopelijk beter resultaten worden gegeven in vergelijking met de vorige iteratie.

Dataverzameling

De classifier kan worden getraind om in elk gewenst domein ingezet te worden. Hierdoor zijn de labels die gekozen moeten worden vrij.

Een mogelijk gebruik van de Natural Language Classifier is om het verwachte type antwoord (lexical answer type) bij een verzekeringsvraag te achterhalen. De labels die hierbij horen zijn bijvoorbeeld:

- coverage;
- definition;
- guidance;
- deductible;
- date;
- value.

Deze labels geven aan welk type antwoord een gebruiker verwacht bij een bepaalde verzekeringsvraag. Bij een vraag geclassificeerd als `coverage` wordt bijvoorbeeld een antwoord verwacht waarin aangegeven wordt of iets wel of niet gedekt wordt door de verzekering. Wanneer de vraag wordt geclassificeerd als `definition` wordt een definitie als antwoord verwacht.

De manier waarop tekst door deze service wordt geclassificeerd is door gebruik te maken van de trainingsdata als voorkennis. De trainingsdata van de Natural Language Classifier service bestaat uit paren van tekst en labels in een CSV-bestand. In figuur 7 hieronder staat een voorbeeld van hoe dit eruit kan zien voor het bovenstaande voorbeeld.

Am I covered for *	, coverage
Is * insured	, coverage
* am I insured for this	, coverage
What does * mean	, definition
What do you mean with *	, definition
*?	, definition
*, what should I do now	, guidance
Can you help me through the * process	, guidance

Figuur 7 Natural Language Classifier service - groundtruth voorbeeld.

De bovenstaande voorbeelden gebruiken een asterisk als wildcard. Dit teken is alleen toegevoegd voor de overzichtelijkheid en kan weggelaten worden. Aangezien patroonherkenning wordt gebruikt om de input te herkennen zorgt het gebruik van de asterisk niet voor problemen met de classificatie. De asterisk staat in vrijwel alle zinnen en heeft daarom weinig invloed op de uitkomst van de classificatie.

Gebruik

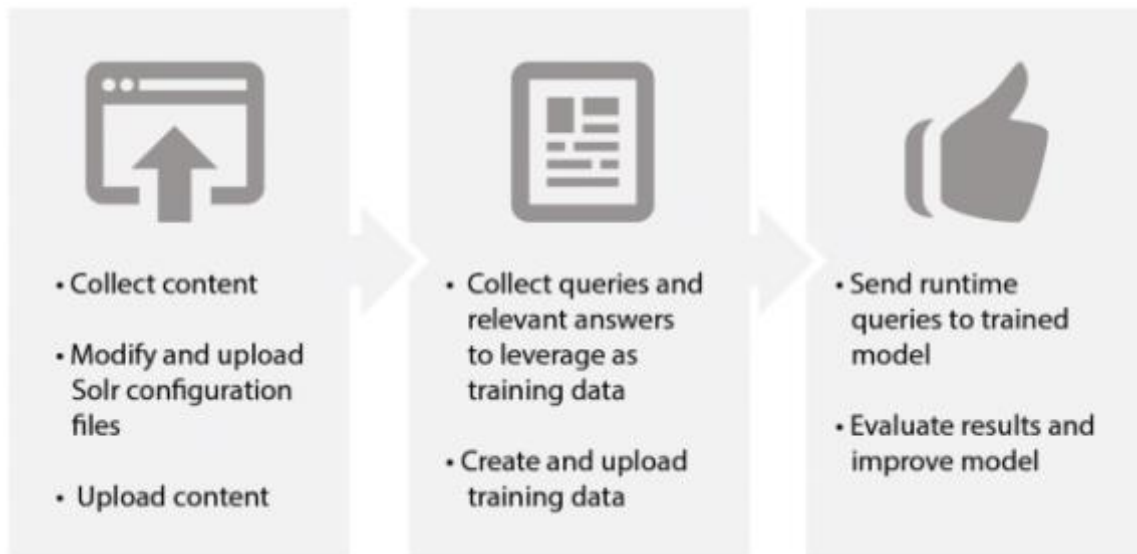
De resultaten van de classifier kunnen bewaard worden om op een later stadium te beoordelen. De input van de resultaten die onjuist zijn geclassificeerd kunnen letterlijk worden toegevoegd aan de trainingsdata om het de volgende iteratie wel juist te laten verlopen. Doordat het systeem werkt met patroonherkenning zal de hoeveelheid trainingsdata ervoor zorgen dat beter wordt geclassificeerd.

8.4 Retrieve and Rank

De Retrieve and Rank service wordt gebruikt als cognitieve zoekmachine om te kunnen citeren uit bijvoorbeeld polisvoorwaarden documenten. De retrieve en rank service bestaat zoals de naam al doet vermoeden uit een retrieve onderdeel en een rank onderdeel. Het retrieve gedeelte is gebouwd op het opensource beschikbare Solr -project. Het rank gedeelte is door IBM toegevoegd om het geheel cognitief te maken.

Workflow

In figuur 8 hieronder wordt aangegeven welke workflow gebruikelijk is bij het implementeren en gebruiken van de Retrieve and Rank service. De derde stap is een iteratieve stap.



Figuur 8 Retrieve and rank service - implementatie workflow (IBM, 2016d)

De eerste stap bestaat uit het verzamelen van de opvraagbare gegevens. Deze data bestaat uit een collectie van answer units zoals bijvoorbeeld de resultaten van de Document Conversion service. Aan deze data wordt desgewenst metadata toegevoegd zoals bijvoorbeeld het type antwoord wat wordt gegeven. Een answer unit is een mogelijk antwoord van de Retrieve and Rank service geformatteerd in JSON-formaat. Aan de hand van de JSON-indeling wordt Solr geconfigureerd.

In stap twee worden mogelijke zoekopdrachten verzameld en deze worden toegewezen aan de relevante answer units uit de in stap één verzamelde data. Met deze verzameling wordt trainingsdata gecreëerd om de ranker mee te maken.

De laatste stap is een iteratieve stap waarin de service gebruikt en verbeterd wordt aan de hand van de verzamelde gebruiksgegevens. Dit is weer hetzelfde principe als bij de Dialog en Natural Language Classifier services.

Dataverzameling

Als eerste wordt de data verzameld. Deze data bestaat bijvoorbeeld uit de verzameling van polisvoorwaarden documenten. De polisdocumenten worden met behulp van de Document Conversion service omgezet in answer units, waarna deze ingelezen kunnen worden door de Retrieve and Rank service. Deze verzameling van answer units wordt ook wel het *corpus* genoemd.

Aan de hand van de answer units wordt Solr geconfigureerd. Het configureren van deze cluster is Solr specifiek en daarom niet opgenomen in dit document. Een standaard configuratie is beschikbaar voor de Solr-instantie. De belangrijkste veranderingen aan deze standaard configuratie is het aangeven van de JSON-formaat. Dit gebeurt in het schema.xml-bestand van de Solr-configuratie. Een voorbeeld is hieronder in figuur 9 gegeven. Bovenin staat een answer unit met daaronder de bijbehorende Solr-configuratie.

```
{
  "id": 2,
  "doc_id": 1,
  "type": "h3",
  "parent_id": 1,
  "lexical_answer_type": "...",
  "title": "...",
  "content": "..."
}
```

```
<field name="id" type="string" required="true"/>
<field name="doc_id" type="string" required="true"/>
<field name="type" type="watson_text_en"/>
<field name="parent_id" type="string"/>
<field name="lexical_answer_type" type="watson_text_en"/>
<field name="title" type="watson_text_en"/>
<field name="content" type="watson_text_en"/>
```

Figuur 9 Retrieve and Rank service - voorbeeld Solr-configuratie

Trainen van de ranker

Voor het trainen van de ranker wordt een "ground truth" CSV-bestand aangemaakt waarin mogelijke vragen worden opgesomd. Aan deze vragen worden paren toegevoegd van id en relevantie. Figuur 10 hieronder geeft hiervan een voorbeeld.

Am I insured for damage to my windshield?,	165,	1,	289,	2
When should I file an auto insurance claim?,	26,	2		
Is my own repair work insured?,	132,	1,	489,	0

Figuur 10 Retrieve and Rank service - groundtruth voorbeeld

In dit voorbeeld heeft de eerste vraag een referentiescore van 1 bij de answer unit met ID 165 en een referentiescore van 2 bij de answer unit met ID 289. De tweede vraag heeft een referentiescore van 2 bij answer unit 26. Hogere scores geven een hogere relevantie aan. De derde vraag heeft een referentiescore van 0 bij answer unit 489. De score 0 is vooraf gedefinieerd en geeft aan dat het bijbehorende answer unit niet als antwoord gegeven mag worden.

Verdere training

Het verder trainen van de Retrieve and Rank service gaat op dezelfde manier als bij de Dialog en Natural Language Classifier services. Het gebruik wordt gelogd waarna deze op een later stadium bekeken en verwerkt kan worden.

8.5 Configuratiemogelijkheden

Alle Watson Bluemix services zijn te benaderen via een REST API. Communicatie met deze services is alleen mogelijk via deze REST API. Aangezien Colimbra de applicatie in C# wil realiseren moet voor het gebruik van de services een wrapper worden gemaakt. De configuratie van deze services kan direct via de REST API gebeuren of door de REST API te verwerken in het prototype.

- **Dialog**
De Dialog service heeft een dialoogscript nodig in XML-formaat waarin het conversatieverloop wordt gedefinieerd. Wanneer dit script is ingelezen wordt tekst via de REST API verstuurd welke op de IBM-servers door het script worden gehaald. Gebruik dient gelogd te worden om in een later stadium het dialoogscript uit te breiden.
- **Document conversion**
De Document Conversion service converteert documenten aan de hand van een in JSON-formaat meegeleverde configuratiefile. Hier zijn geen cognitieve functies aan toegevoegd.
- **Natural language classifier**
De Natural Language Classifier service biedt de mogelijkheid om meerdere classificatiesystemen aan te maken. Per classificatiesysteem wordt trainingsdata in CSV-formaat gegeven waarna deze data als basiskennis wordt gebruikt om input te classificeren. Het gebruik wordt hier weer gelogd waarna de trainingsdata aan gevuld kan worden in een later stadium.
- **Retrieve and rank**
De Retrieve and Rank service heeft meer configuratie nodig dan de voorgaande services. De Retrieve and Rank service maakt gebruik van een interne Solr-instantie welke ingesteld moet worden om de gewenste answer units te kunnen verwerken. Na instellen van de Solr-cluster moet het corpus opgebouwd worden. Dit is een verzameling van answer units uit verschillende bronnen. Na het verzamelen van de data dient de trainingsdata worden gemaakt. Deze trainingsdata bestaat uit een verzameling van vragen gevolgd door paren van ID en relevantiescore. De trainingsdata kan net zoals de Dialog en Natural Language Classifier services worden aangevuld aan de hand van gelogd gebruik.

9. Vraag- en antwoordsysteem

In dit hoofdstuk wordt de verzamelde informatie uit de vorige hoofdstukken gebruikt om het benodigde vraag- en antwoordsysteem op te kunnen zetten. De modellen in dit hoofdstuk zijn ook opgenomen in het technische ontwerp, toegevoegd als bijlage C. De architectuur in dit hoofdstuk is in stand gekomen aan de hand van een gesprek met (Sips, 2016) van IBM.

Dit hoofdstuk geeft een antwoord op deelvraag 5:

Hoe kunnen de relevante services aan elkaar geknoopt worden om een vraag- en antwoordsysteem te realiseren?

In hoofdstuk 8 is aangegeven waarvoor de verschillende services dienen en hoe deze gebruikt kunnen worden. Om een vraag- en antwoordsysteem op te zetten zullen de relevante services met elkaar moeten samenwerken. De onderstaande onderdelen worden besproken in dit hoofdstuk.

- **Architectuur**

Het vraag- en antwoordsysteem bestaat uit de vier relevante services besproken in de vorige paragraaf. Het systeem dat op wordt gezet voor het prototype maakt gebruik van de Natural Language Classifier en Dialog services om vaak gestelde vragen te herkennen en beantwoorden. De Retrieve and Rank service wordt gebruikt om de vragen die niet vooraf gedefinieerd zijn te kunnen beantwoorden. De werking hiervan wordt besproken in paragraaf 9.1.

- **Modules**

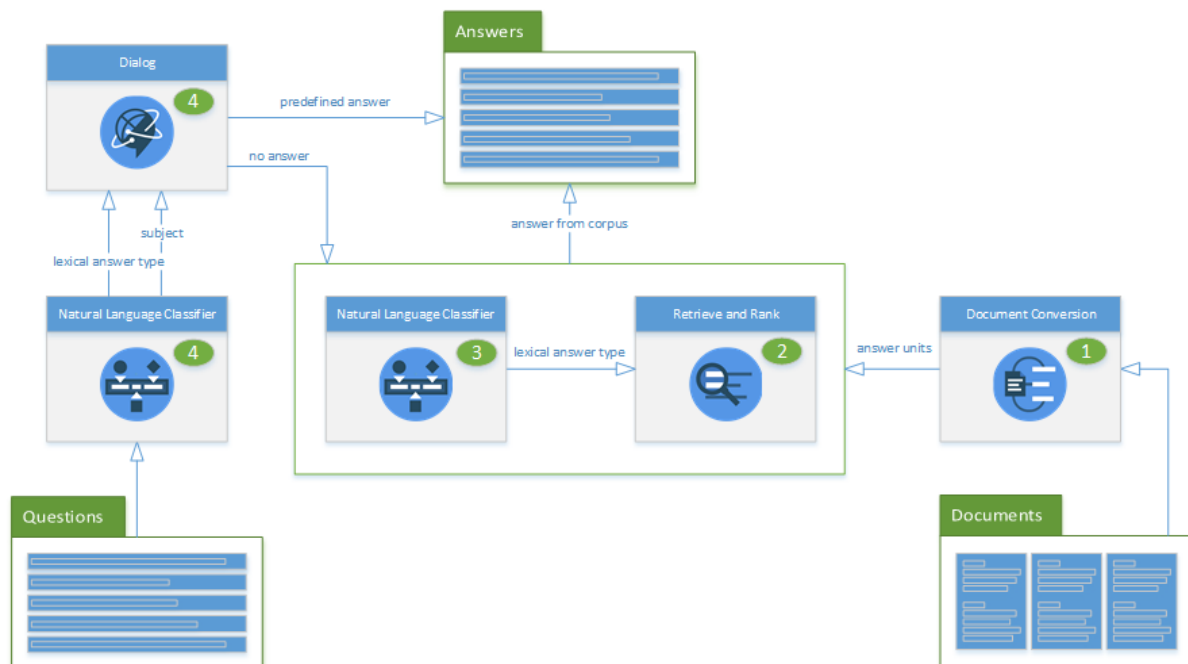
In het vorige kopje is aangegeven dat de Retrieve and Rank service wordt gebruikt om de vragen te beantwoorden. Voor het prototype wordt deze service gevuld met data opgesteld door domeinexperts. Er kunnen echter meerdere databronnen gebruikt worden welke beschouwd worden als verschillende modules. Zo kan bijvoorbeeld een nieuwe Retrieve and Rank cluster gebruikt worden om vragen te kunnen beantwoorden met behulp van data uit Wikipedia. Een aantal mogelijke modules worden beschreven in paragraaf 9.2.

Prototype

Voor het prototype zijn geen grote hoeveelheden trainingsdata beschikbaar. Hierdoor is besloten om de onderstaande architectuur te gebruiken. Uitleg over deze architectuur inclusief beargumentering wordt gegeven in paragraaf 9.3.

9.1 Architectuur

De Dialog, Document Conversion, Natural Language Classifier, en Retrieve and Rank services kunnen gebruikt worden om een vraag- en antwoordsysteem op te zetten. Figuur 11 geeft aan op welke manier dit gerealiseerd kan worden.



Figuur 11 Architectuurdiagram - kern

Zoals aangegeven in figuur 11 is de eerste stap het verzamelen van de opvraagbare gegevens. Deze gegevens worden met behulp van de Document Conversion service omgezet in een verzameling van answer units. Desgewenst kunnen de gegenereerde answer units worden aangevuld met metadata.

De gegenereerde answer units worden ingelezen in de Retrieve and Rank service. Deze service wordt gebruikt om de zogenaamde "long tail" van vragen te beantwoorden. Dit zijn de vragen waarop geen vooraf gedefinieerd antwoord gevonden is. Na het inlezen van deze documenten dient de Retrieve and Rank service getraind te worden om vragen te kunnen beantwoorden.

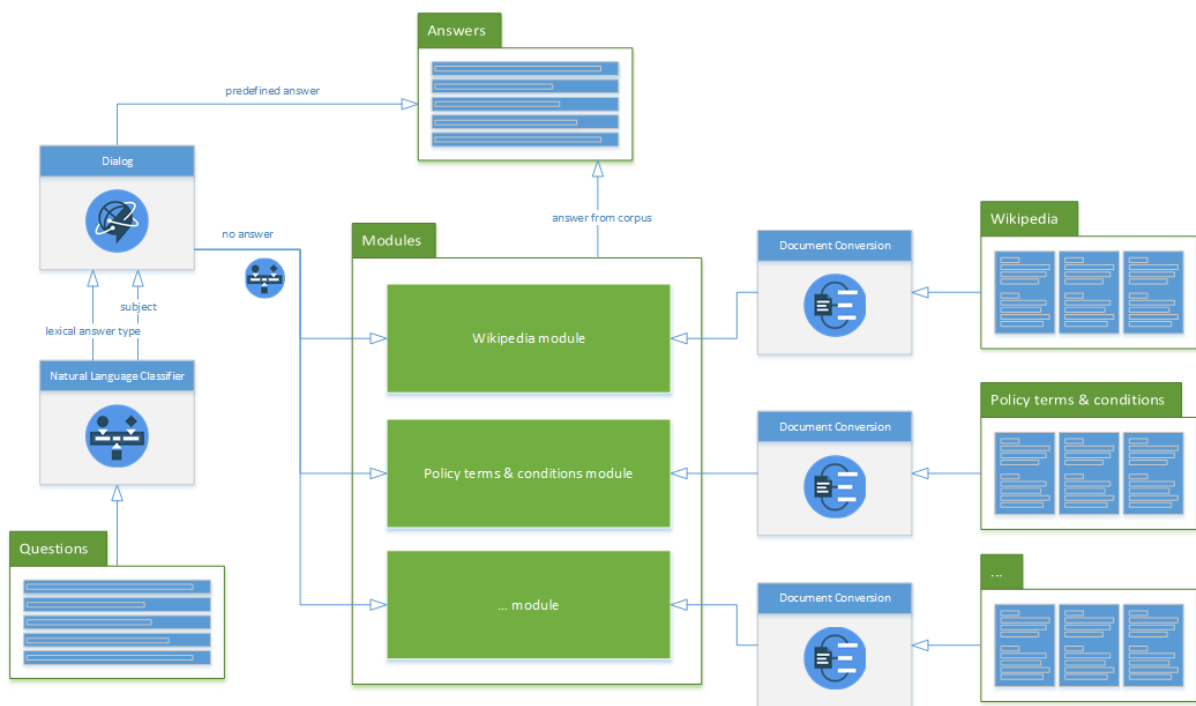
Nadat de Retrieve and Rank service is getraind kan de Natural Language Classifier service ingezet worden om te achterhalen wat voor soort antwoord de vragende gebruiker verwacht. Dit wordt de "lexical answer type" genoemd. Voor efficiëntie kunnen de answer units uit de Retrieve and Rank service vooraf aangevuld worden met het type antwoord dat de answer unit bevat. Een andere optie is om dit te classificeren na het opvragen van de answer units. De geclassificeerde lexical answer type kan naast de gevonden answer units gezet worden om zo het resultaat te verbeteren.

Na het toevoegen van de Natural Language Classifier wordt een dialoog gerealiseerd. Dit dialoog gebruikt twee classifiers om een vraag te herkennen. Een nieuwe classifier moet hierdoor aangemaakt worden waarmee het onderwerp van de vraag achterhaald kan worden. De tweede classifier is de lexical answer type classifier aangemaakt in de vorige stap. Met behulp van de output van deze twee classifiers kunnen vragen, die regelmatig voorkomen, herkend worden. Het antwoord op deze vragen wordt verwerkt in het dialoogscript.

9.2 Modules

Het schema in figuur 11 geeft de architectuur aan van een mogelijk vraag- en antwoordsysteem. De combinatie van Natural Language Classifier en Retrieve and Rank services vormen samen één module die gebruikt kan worden om informatie op te halen. Meerdere van dit soort modules kunnen gebruikt worden om over verschillende databronnen te beschikken.

Wanneer meerdere modules worden gebruikt, kan gekozen worden om een nieuwe classifier te gebruiken waarmee besloten wordt welke module gebruikt wordt voor het beantwoorden van een bepaalde vraag. Ook kan gekozen worden om alle modules te gebruiken en aan de hand van de zekerheidsgraad een beslissing te maken over het antwoord dat uiteindelijk gegeven wordt. Deze architectuur wordt hieronder getoond in figuur 12.



Figuur 12 Architectuur diagram - uitgebreid

Hieronder en op de volgende pagina staan een aantal modules die mogelijk gebruikt kunnen worden om de resultaten van Watson te verbeteren.

Polisvoorwaarden documenten

Voor de verzekeringsbranche is een belangrijke module de polisvoorwaardenmodule. Deze module bestaat uit de Natural Language Classifier en Retrieve and Rank services om informatie te vinden uit polisdocumenten.

Wikipedia

De Watson architectuur kan uitgebreid worden om Wikipediapagina's te gebruiken als bron van informatie. De Document Conversion service accepteert HTML-pagina's als input. Op dezelfde manier waarop PDF-documenten worden toegevoegd kunnen ook Wikipediapagina's worden toegevoegd. Dit kan handig zijn om algemene vragen te beantwoorden zoals bijvoorbeeld "Wat is een verzekering?". Het antwoord hierop zal niet te vinden zijn in polisvoorwaardendocumenten aangezien het antwoord op deze vraag gezien wordt als basiskennis. Het antwoord kan wel gevonden worden op Wikipedia.

Helpdeskgesprekken

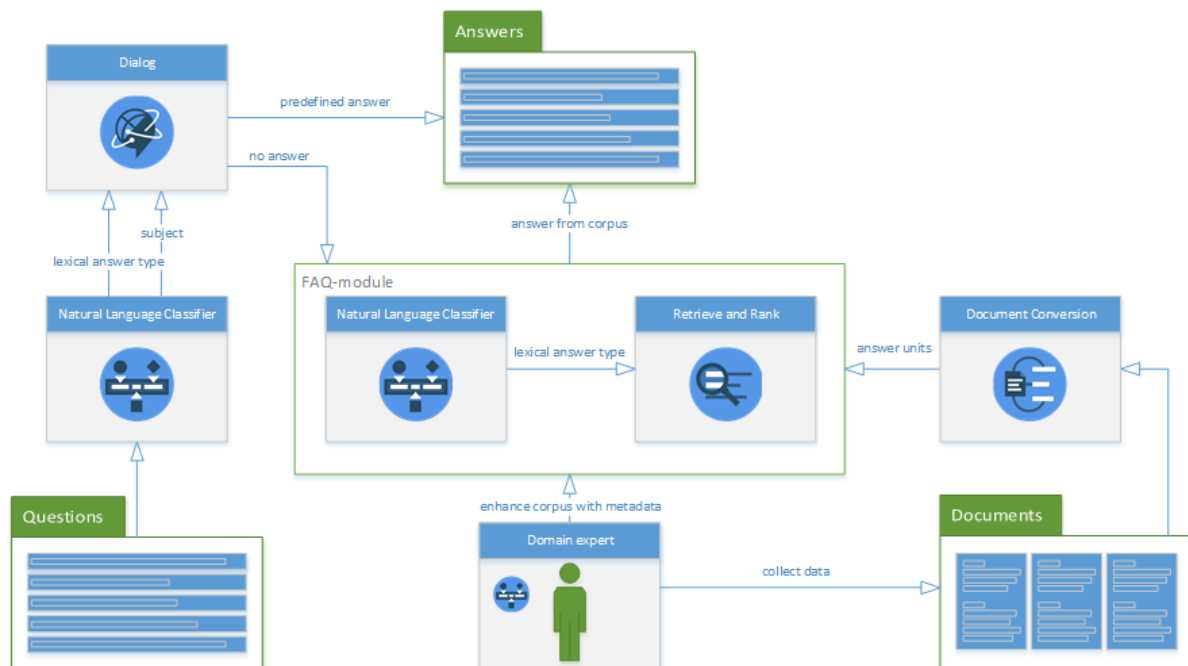
Een grote bron van data kan afkomstig zijn uit helpdeskgesprekken. Om deze reden kan gekozen worden om een verzameling van gesprekken te verwerken in een module. Deze module kan gebruik maken van de speech to text service om de gesprekken om te zetten naar tekst.

Social media

Meer informatie kan verzameld worden door gebruik te maken van social media. Het gaat hierbij waarschijnlijk om vragen waarbij de status van een bepaald proces wordt opgehaald/bijgehouden. Een vraag als "Klopt het dat de website van [verzekeraar] uit de lucht is?" kan met behulp van een social media module beantwoord worden. Het antwoord kan letterlijk afkomstig zijn uit de officiële social media pagina van het bedrijf of er kan geredeneerd worden aan de hand van het aantal klachten. Een social media module zal waarschijnlijk uit meer services bestaan dan alleen de Natural Language Classifier en de Retrieve and Rank services. De tone analyzer en de personality insights services, beschreven in bijlage D, kunnen deze module verbeteren.

9.3 Prototype

Het prototype zal gebruik maken van de architectuur getoond in figuur 13 hieronder.



Figuur 13 Architectuurdiagram - prototype

Hier wordt één module in verwerkt aangezien voor dit prototype geen grote hoeveelheid trainingsdata beschikbaar is. De module die verwerkt wordt bestaat uit een handmatig opgestelde collectie van vraag en antwoord paren.

De gegenereerde answer units worden vooraf aangevuld met de lexical answer type van het antwoord. Voor dit prototype gebeurt dat handmatig, in principe kan zoals in paragraaf 9.1 is aangegeven ook een classifier gebruikt worden om dit te doen.

Deze module zal in een release versie verwerkt moeten worden in de Dialog service aangezien deze vragen vallen onder de vaak gestelde vragen. Voor het prototype wordt dit echter ingeladen als module om de werking van de Retrieve and Rank service te toetsen.

10. Overige benodigdheden

De services van Watson bieden de mogelijkheid om een cognitief vraag- en antwoordsysteem op te zetten. Buiten dit vraag- en antwoordsysteem zijn een aantal andere onderdelen nodig. De overige benodigdheden en de oplossingen die hiervoor zijn gekozen worden beschreven binnen hoofdstuk.

Dit hoofdstuk geeft een antwoord op deelvraag 6:

Welke extra functionaliteit, niet ondersteund door IBM Watson, moet ontwikkeld worden en hoe wordt dit gedaan?

De onderstaande onderdelen worden niet geïmplementeerd met behulp van Watson services. In het verloop van dit hoofdstuk zullen de hieronder genoemde onderdelen worden beschreven.

- **Interface**

Het prototype zal gepresenteerd worden op een verzekeringsbeurs. Hierdoor zal dit prototype een fatsoenlijke interface moeten hebben. In een eerder stadium is besloten dat de focus ligt op de werking van Watson en niet op het maken van een interface. Hierdoor is besloten om het vraag- en antwoordsysteem te verwerken in een bestaande chatapplicatie: "Telegram".

Telegram is een chatapplicatie vergelijkbaar met bijvoorbeeld de bekendere "WhatsApp" applicatie. Voor Telegram is gekozen omdat Telegram een REST API aanbiedt waarmee chatbots geïmplementeerd kunnen worden, dit maakt het mogelijk om Watson te integreren in Telegram en zo tijd te besparen op het ontwikkelen en ontwerpen van een interface. De interfacekeuze wordt in meer detail besproken in paragraaf 10.1.

- **Database**

De trainingsdata van het prototype moet uitgebreid kunnen worden met ervaringen die de applicatie opdoet. Dit betekent dat een database (MongoDB) opgezet moet worden waarin de interactie met gebruikers wordt bijgehouden. Deze database kan ook gevuld worden met persoonsgegevens om bijvoorbeeld toekomstige vragen te kunnen beantwoorden met eerder verzamelde gegevens. Er is gekozen voor MongoDB omdat Colimbra bekend is met deze database en al een aantal applicaties in productie heeft staan welke gebruik maken van MongoDB. De databaseimplementatie wordt beschreven in paragraaf 10.2.

10.1 Interface

Voor de implementatie van de interface is gekozen om een bestaande interface te gebruiken. Hiervoor is gekozen omdat het onderzoek vooral op de werking van Watson moet focussen en niet op het implementeren en ontwerpen van een interface.

Colimbra Application Solution

Voor de realisatie van een interface zijn twee oplossingen overwogen. Allereerst is overwogen om de bestaande ontwikkelsuite van Colimbra te gebruiken, de Colimbra Application Solution (hierna CAS). CAS wordt intern gebruikt om webapplicaties mee te ontwikkelen. De ontwikkelaars van Colimbra kunnen in CAS snel een interface maken om te gebruiken voor het prototype.

Telegram

Uiteindelijk is gekozen om het vraag- en antwoordsysteem te verwerken in Telegram met behulp van een chatbot. Zoals in de inleiding van dit hoofdstuk al is aangegeven is Telegram een chatapplicatie vergelijkbaar met de beter bekende "WhatsApp" applicatie. Voor Telegram is gekozen omdat Telegram het mogelijk maakt chatbots te implementeren waar gebruikers mee kunnen communiceren. Hierdoor kan het vraag- en antwoordsysteem verwerkt worden in een Telegram chatbot. Een andere reden waarom voor Telegram is gekozen is omdat de afstudeerder voor de start van de afstudeerperiode al ervaring heeft opgedaan met de Telegram bot API's. Hierdoor blijft de focus van het onderzoek op de werking van Watson.

Door gebruik te maken van Telegram hoeft geen interface ontworpen te worden. Gebruikers kunnen met behulp van de bestaande Telegram interfaces communiceren met een chatbot. De chatbot dient hier als doorgeefluik naar het vraag- en antwoordsysteem.

De werking van de Telegram bot API's is niet onderzocht gedurende de afstudeerperiode. Daarom zal dit niet uitgebreid besproken worden. Kort samengevat zijn er twee manieren waarop een chatbot data kan ontvangen vanuit Telegram. Dit kan met behulp van longpolling of door een webhook te registreren. In beide gevallen worden updates in JSON-formaat binnen gehaald. Deze updates beschikken over voldoende informatie om een reactie terug te kunnen sturen.

Meer informatie over de Telegram bot API's is te vinden op de officiële Telegram website (Telegram, 2016). Een beschrijving van de Telegram bot implementatie wordt gegeven in paragraaf 12.3.

10.2 Database

De database die gebruikt wordt is MongoDB. Voor deze database is gekozen omdat Colimbra bekend is met de werking van deze database. MongoDB is een NoSQL database om JSON-geformatteerde data in op te slaan (MongoDB, 2016).

Omdat het ontwerpen van een database veel tijd in beslag kan nemen wordt deze pas ontworpen wanneer de database daadwerkelijk geïmplementeerd wordt in het prototype. Hiervoor is gekozen omdat in de MoSCoW methode uit paragraaf 6.2 (figuur 3) aangegeven wordt dat de database implementatie een "Could have" betreft en dus geen hoge prioriteit heeft.

11. Training

De Dialog, de Natural Language Classifier en de Retrieve and Rank services moeten getraind worden voordat deze gebruikt kunnen worden. Dit hoofdstuk beschrijft welke trainingsdata nodig is om kennis op te doen over de verzekeringsbranche. Alle informatie uit deze paragraaf is afkomstig uit de officiële IBM Watson documentatie (IBM, 2016d).

Dit hoofdstuk geeft een antwoord op deelvraag 7:

In hoeverre dient IBM Watson getraind te worden om kennis op te doen over de verzekeringsbranche?

De informatie in dit hoofdstuk is deels al gegeven in de serviceomschrijving in hoofdstuk 7. Dit hoofdstuk geeft buiten de benodigde trainingsdata ook aan welke eisen gesteld worden aan de desbetreffende trainingsdata en welke bronnen gebruikt kunnen worden om de trainingsdata op te stellen.

Voor de services beschreven in hoofdstuk 8 zal aangeven welke eisen worden gesteld aan de benodigde trainingsdata. De Document Conversion service wordt hier niet in mee genomen omdat deze service geen gebruik maakt van trainingsdata.

Dialog

De Dialog service heeft zoals eerder is aangegeven een script nodig waarin het conversatieverloop vooraf wordt gedefinieerd. Dit script wordt door IBM beschouwd als de trainingsdata voor de Dialog service. De enige eisen aan dit script zijn afkomstig uit het gebruikte XML-schema zoals kort besproken in paragraaf 8.1.

Natural Language Classifier

De Natural Language Classifier service werkt door gebruik te maken van patroonherkenning. Zoals beschreven in paragraaf 8.3 bestaat de trainingsdata uit een CSV-bestand. Deze service heeft geen minimale hoeveelheid trainingsdata nodig.

Retrieve and Rank

De Retrieve and Rank service heeft het meeste trainingsdata nodig. In paragraaf 8.4 is beschreven dat de trainingsdata bestaat uit een CSV-bestand waarin vragen worden opgesomd met achter deze vragen paren van id en relevantie.

Van de trainingsdata wordt verwacht dat het aan de volgende kwaliteitscriteria voldoet:

- Het CSV-bestand moet op z'n minst 49 unieke vragen bevatten;
- Het aantal answer units moet op z'n minst 50 keer het aantal velden, geconfigureerd in de Solr-configuratie, groot zijn.
- Er moeten op z'n minst twee verschillende referentielabels gebruikt worden.
- Het getal 0 wordt niet gebruikt op de referentieschaal. Dit getal geeft aan dat een bepaalde answer unit juist niet gerefereerd mag worden.

12. Prototype

Dit hoofdstuk beschrijft het gerealiseerde prototype. Het prototype dat beschreven is tijdens het onderzoek (paragraaf 6.2) is uiteindelijk deels gerealiseerd. De Must en de Should onderdelen uit de MoSCoW methode van figuur 3 zijn verwerkt in dit prototype. In dit hoofdstuk wordt de technische werking van het prototype beschreven. De modellen in dit hoofdstuk zijn ook verwerkt in het technisch ontwerp, toegevoegd als bijlage C.

Er worden REST API's gebruikt om Watson en Telegram te benaderen. Voor beide API's zijn in C# wrappers aangemaakt. Voor de Watson implementatie is een WatsonCore project aangemaakt. De Telegram Bot implementatie is gerealiseerd in een ColimbraBot project.

- **WatsonAPI**
De Watson API's bieden een wrapper om de REST API's van de Dialog, de Natural Language Classifier, de Document Conversion en de Retrieve and Rank services. Informatie over de implementatie hiervan wordt gegeven in paragraaf 12.1.
- **WatsonCore**
Het WatsonCore project bevat de implementatie van Watson. Dit project maakt gebruik van het WatsonAPI project om te communiceren met de Watson services. Dit project biedt een Watson instantie waarmee gecommuniceerd wordt. Een gedetailleerdere beschrijving van dit project wordt gegeven in paragraaf 12.2.
- **TelegramAPI**
De Telegram API is ontwikkeld als C# wrapper om de REST API voor Telegram Bots. Een abstracte klasse wordt beschikbaar gesteld die gebruikt kan worden om een Telegram Bot te implementeren. De manier waarop deze wrapper gebruikt kan worden is beschreven in paragraaf 12.3.
- **ColimbraBot**
De communicatie met Watson verloopt zoals hiervoor vermeld met behulp van een Telegram Bot. Dit project is het startup project waarmee Watson gestart wordt. De TelegramAPI wordt gebruikt om de communicatie met de Telegram REST API te realiseren en het WatsonCore project wordt gebruikt om de ontvangen tekst door te sturen naar de Watson implementatie. Een gedetailleerdere beschrijving van dit project wordt gegeven in paragraaf 12.4.

De bovengenoemde services zijn geïmplementeerd met behulp van het .NET core framework. Dit is een relatief nieuw framework van Microsoft. Voor dit framework is gekozen omdat dit framework cross-platform is en vaak gebruikte services ingebouwd heeft zoals bijvoorbeeld een dependency injection (DI) container en een implementatie voor configuratiefiles (.Net Foundation, 2016). Overigens is dit framework open-source.

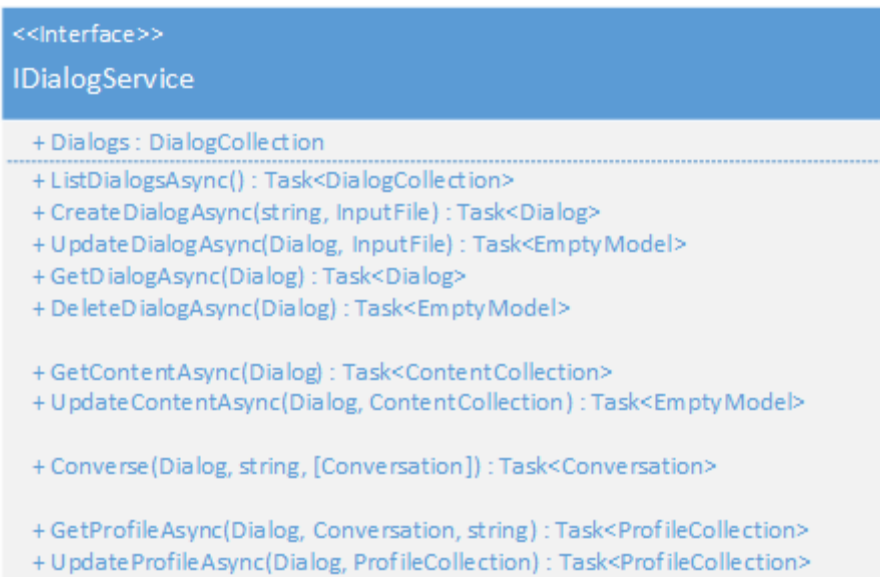
12.1 WatsonAPI

Het Watson API-project biedt een C# wrapper aan voor de REST API's van de Dialog, de Document Conversion, de Natural Language Classifier en de Retrieve and Rank services. Dit project bevindt zich in de `Colimbra.Watson.Api` namespace.

Het WatsonAPI project implementeert de onderstaande services. Voor elk van de onderstaande services is een extensiemethode toegevoegd aan de `IServiceCollection` klasse om deze toe te kunnen voegen aan de DI-container. Dit gebeurt op een singleton manier, alleen wanneer de desbetreffende service nog niet aanwezig is. Bij elke service is een deel van het klassendiagram gegeven waarin te zien is welke functionaliteit deze service biedt.

Dialog

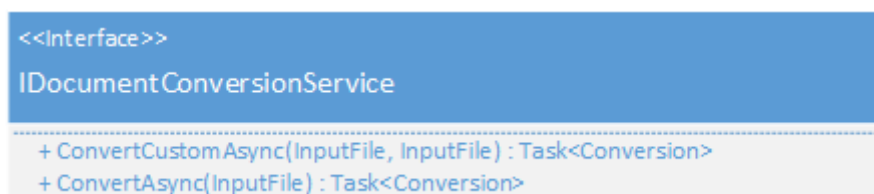
De `IDialogService` service is geïmplementeerd in de `DialogService` klasse. Deze service kan aan de DI-container toegevoegd worden met behulp van de `IServiceCollection.AddDialog()` extensiemethode.



Figuur 14 Dialog service

Document conversion

De `IDocumentConversionService` service is geïmplementeerd in de `DocumentConversionService` klasse. Deze service kan toegevoegd worden aan de DI-container met behulp van de `IServiceCollection.AddDocumentConversion()` extensiemethode.



Figuur 15 Document Conversion service

Natural Language Classifier

De `NaturalLanguageClassifierService` klasse wordt gebruikt als basisimplementatie voor de `INaturalLanguageClassifierService` service. Het toevoegen van deze service aan de DI-container gebeurt met de `IServiceCollection.AddNaturalLanguageClassifier()` extensiemethode.

```
<<Interface>>
INaturalLanguageClassifierService

+ Classifiers : ClassifierCollection
+ ListClassifiersAsync() : Task<ClassifierCollection>
+ CreateClassifierAsync(string, InputFile) : Task<Classifier>
+ GetClassifierAsync(Classifier) : Task<Classifier>
+ DeleteClassifierAsync(Classifier) : Task<Classifier>

+ ClassifyAsync(Classifier, string) : Task<Classification>
```

Figuur 16 Natural Language Classifier service

Retrieve and Rank

De `IRetrieveAndRankService` service is geïmplementeerd in de `RetrieveAndRankService` klasse. Deze service kan toegevoegd worden aan de DI-container met behulp van de `IServiceCollection.AddRetrieveAndRank()` extensiemethode.

```
<<Interface>>
IRetrieveAndRankService

+ Clusters : ClusterCollection
+ Rankers : RankerCollection
+ ListClustersAsync() : Task<ClusterCollection>
+ CreateClusterAsync(string, string) : Task<Cluster>
+ GetClusterAsync(Cluster) : Task<Cluster>
+ DeleteClusterAsync(Cluster) : Task<EmptyModel>

+ ListConfigurationsAsync(Cluster) : Task<ConfigurationCollection>
+ CreateConfigurationAsync(Cluster, string, InputFile) : Task<Configuration>
+ GetConfigurationAsync(Cluster, Configuration) : Task<Configuration>
+ DeleteConfigurationAsync(Cluster, string) : Task<EmptyModel>

+ ListCollectionsAsync(Cluster) : Task<CollectionCollection>
+ CreateCollectionAsync(Cluster, Configuration, string) : Task<EmptyModel>
+ DeleteCollectionAsync(Cluster, string) : Task<EmptyModel>

+ SelectAsync(Cluster, string, string, [List<string>]) : Task<SolrResponse>
+ FcSelectAsync(Cluster, string, Ranker, string, [List<string>]): Task<SolrReponse>

+ ListRankersAsync() : Task<RankerCollection>
+ CreateRankerAsync(Cluster, string, string, InputFile) : Task<Ranker>
+ GetRankerAsync(Ranker) : Task<Ranker>
+ DeleteRankerAsync(Ranker) : Task<EmptyModel>
+ RankAsync(Ranker, InputFile) : Task<Rank>
```

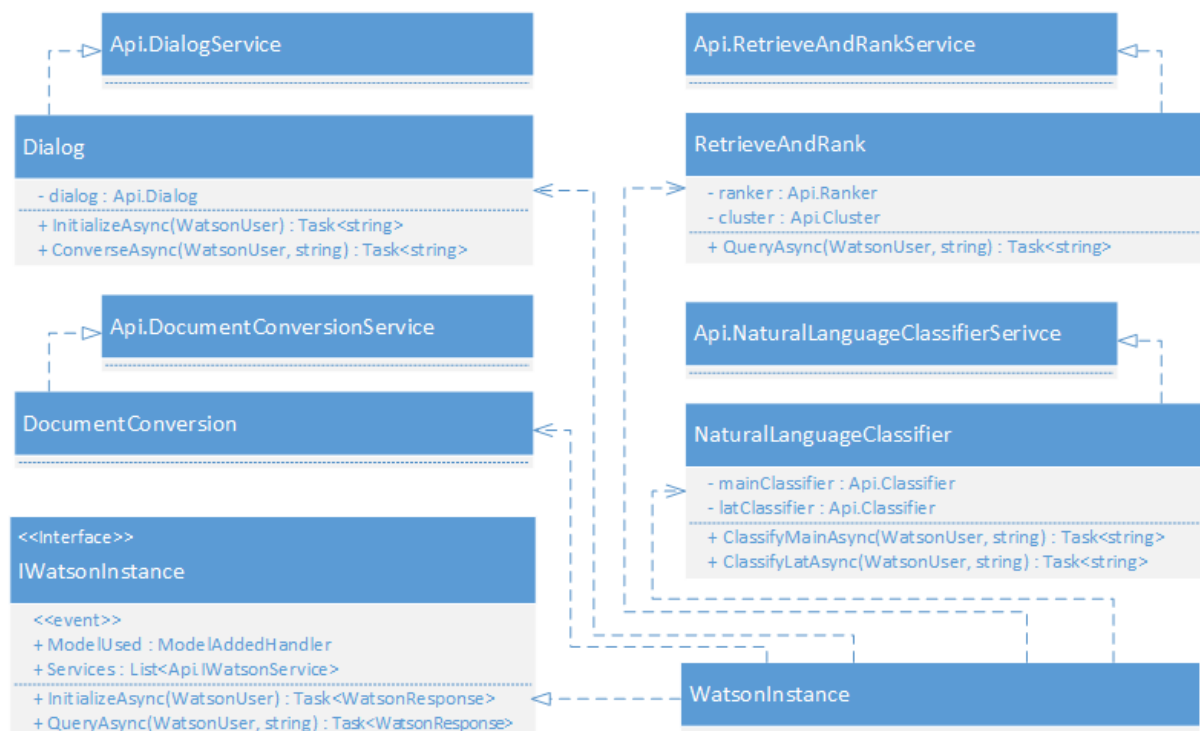
Figuur 17 Retrieve and Rank service

12.2 WatsonCore

Het WatsonCore project is afhankelijk van het WatsonAPI project voor communicatie met de Watson services. Het WatsonCore project bevat de Watson implementatie voor Colimbra. Een project dat gebruik wil maken van de Watson implementatie hoeft alleen te communiceren met de `IWatsonInstance` service. Dit project bevindt zich in de `Colimbra.Watson.Core` namespace.

Watson instantie

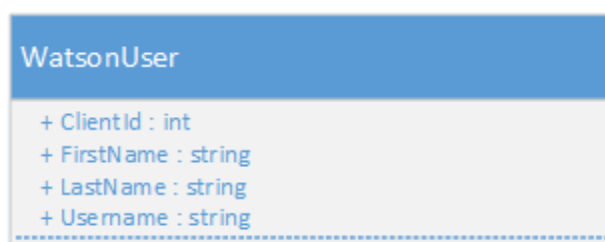
De `WatsonInstance` klasse representeert de Watson implementatie voor Colimbra. Deze service is gedefinieerd in de `IWatsonInstance` interface en kan toegevoegd worden aan een DI-container door de `IServiceCollection.AddWatson()` extensiemethode te gebruiken. Deze methode zorgt ervoor dat een singleton instantie van de `WatsonInstance` inclusief de benodigde Watson services worden toegevoegd.



Figuur 18 Watson service

Watson user

De `WatsonUser` klasse is een eenvoudige POCO-klasse. Deze klasse representeert de gegevens die per persoon door Watson worden bijgehouden. Tijdens de initialisatie en communicatie met Watson moet een `WatsonUser` worden meegegeven. Tijdens initialisatie zal, als deze niet al aanwezig was een `ClientId` worden toegewezen (afkomstig uit de Dialog service)



Figuur 19 Watson gebruiker

Configuratie

De `IServiceCollection.ConfigureWatson(IConfiguration)` extensiemethode kan gebruikt worden om een configuratie mee te geven aan de Watson services. Het schema van deze configuratie wordt in figuur 20 getoond aan de hand van een voorbeeld in JSON-formaat.

```
{
  "Dialog": {
    "Username": "USERNAME",
    "Password": "PASSWORD",

    "DialogName": "dialog"
  },

  "DocumentConversion": {
    "Username": "USERNAME",
    "Password": "PASSWORD"
  },

  "NaturalLanguageClassifier": {
    "Username": "USERNAME",
    "Password": "PASSWORD",

    "MainClassifierName": "main_classifier",
    "MainClassifierRequiredConfidence": 0.75,

    "LatClassifierName": "lat_classifier",
    "LatClassifierRequiredConfidence": 0.75
  },

  "RetrieveAndRank": {
    "Username": "USERNAME",
    "Password": "PASSWORD",

    "ClusterName": "cluster",
    "RankerName": "ranker"
  }
}
```

Figuur 20 Watson Core configuratie

Zoals te zien is wordt per service een gebruikersnaam en wachtwoord meegegeven aan de configuratie.

De Dialog service heeft als extra configuratie de mogelijkheid om de naam van het gebruikte dialoogsript aan te geven (deze is bepaald bij het opsturen van het dialoogsript).

De Natural Language Classifier service gebruikt twee classifiers, hiervan kunnen de namen aangegeven worden. Buiten de namen van deze classifiers kan ook per classifier de minimale zekerheidsgraad ingesteld worden.

De Retrieve and Rank service gebruikt een cluster en een ranker, ook hiervan kunnen de namen aangegeven worden.

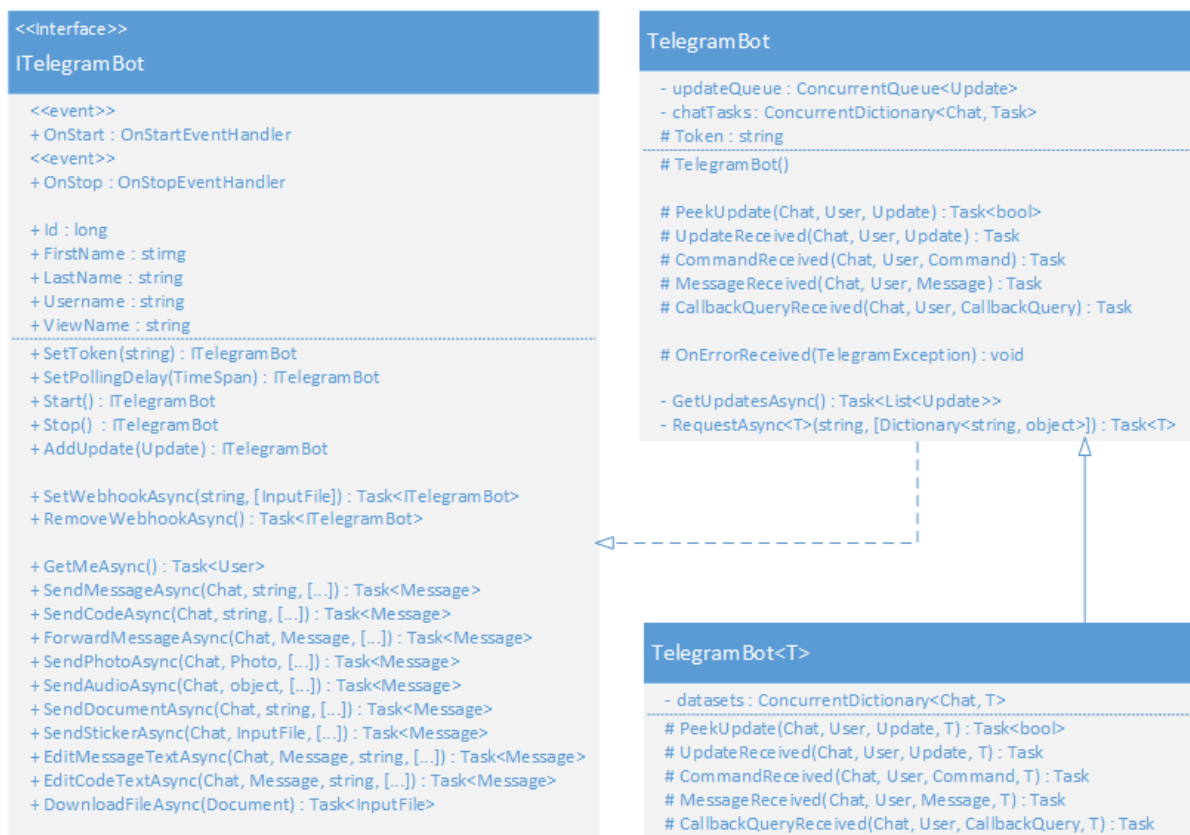
12.3 TelegramAPI

Het TelegramAPI project biedt een C# wrapper aan voor de REST API's voor Telegram bots. Dit project bevindt zich in de `Colimbra.Telegram.Api` namespace.

Telegram bot

De `TelegramBot` klasse wordt gebruikt als abstract basis klasse voor een `ITelegramBot` service.

Figuur 21 weergeeft de `TelegramBot` klasse, deze figuur is voor de overzichtelijkheid niet volledig uitgewerkt.



Figuur 21 TelegramBot service

De `PeekUpdate` methode is bedoeld als check, deze methode wordt als eerste aangeroepen wanneer een bericht vanuit Telegram binnenkomt. Deze methode kan overgeërfd worden om te dienen als filter voor bijvoorbeeld autorisatiechecks.

Zoals in figuur 21 te zien is bevat de `TelegramBot` klasse een aantal `-Received` methoden. Deze methoden kunnen overgeërfd worden om de desbetreffende berichten te verwerken.

De `TelegramBot` klasse is ontworpen om elke chat asynchroon te behandelen. Hierdoor worden, binnen een chat, berichten synchroon behandeld. De ene chat hoeft echter niet op de andere chat te wachten voordat zijn berichten worden afgehandeld.

Zoals in de figuur 21 te zien is, is er ook een `TelegramBot<T>` implementatie. Het type dat meegegeven wordt aan deze klasse wordt gebruikt als dataklasse per chat. Elke chat kan hierdoor zijn eigen instantie van die dataklasse beheren.

12.4 ColimbraBot

Het ColimbraBot project implementeert de Telegram bot. Dit is voor het prototype ook het startup project. Centraal binnen dit project staat de ColimbraBot klasse. Deze wordt in de figuur hieronder getoond.

ColimbraBot : TelegramBot<ColimbraBotData>	ColimbraBotData
- chatAuthorization : List<long> - commandAuthorization : List<long> - developmentMode : bool # PeekUpdate(Chat, User, Update, ColimbraBotData) : Task<bool> # UpdateReceived(Chat, User, Update, ColimbraBotData) : Task # CommandReceived(Chat, User, Command, ColimbraBotData) : Task # MessageReceived(Chat, User, Message, ColimbraBotData) : Task # CallbackQueryReceived(Chat, User, CallbackQuery, ColimbraBotData) : Task - InitializeAsync(Chat, User, ColimbraBotData) : Task	+ WatsonUser : Watson.Core.WatsonUser + LoadingMessage : Telegram.Api.Message + Initialized : bool + ShowJson : bool + ShowSummary : bool + TrainMode : bool + RequireParameter : bool + QueuedCommand : Telegram.Api.Command + QueuedService : Watson.Api.IWatsonService + QueuedMethod : System.Reflection.MethodInfo + RequiredParameters : List<System.Reflection.ParameterInfo> + Parameters : List<object>

Figuur 22 ColimbraBot service

De ColimbraBot klasse is een TelegramBot met de ColimbraBotData klasse als persoonlijke dataset. Dit houdt in dat elke chat een eigen instantie heeft van de ColimbraBotData klasse (zie paragraaf 12.3).

Watson communicatie

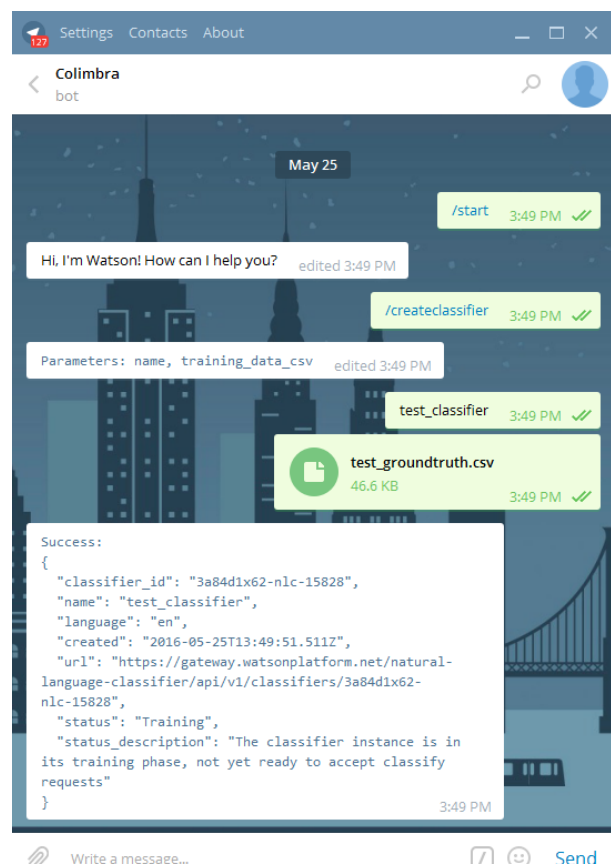
De ColimbraBot is in eerste instantie geïmplementeerd als doorgeefluik naar het WatsonCore project. Hierdoor kan met behulp van de interfaces van Telegram gecommuniceerd worden met Watson. De manier waarop dit gebeurt is door de tekstberichten vanuit de MessageReceived methode door te sturen naar de WatsonInstance klasse.

Watson training

Ook de Watson API's zijn verwerkt in de ColimbraBot. Om dit te realiseren moest elke aanroepbare methode gelabeld worden. De services uit het WatsonCore project zijn overgenomen en elke aanroepbare methode is gelabeld.

Het aanroepen van de Watson API is mogelijk door gebruik te maken van Telegram commando's. Dit zijn tekstberichten die starten met een slash.

Figuur 23 laat een voorbeeld zien voor het aanmaken van een classifier op de desktopinterface. Meerdere voorbeelden van de interfaces zijn toegevoegd aan bijlage E.



Figuur 23 ColimbraBot commando voorbeeld

Configuratie

De `IServiceCollection.ConfigureColimbraBot(IConfiguration)` extensiemethode kan gebruikt worden om een configuratie mee te geven aan de `ColimbraBot` implementatie. Het schema van deze configuratie wordt hieronder getoond aan de hand van een voorbeeld in JSON-formaat.

```
{
  "Watson": { ... },

  "TelegramBot": {
    "Token": "TOKEN",

    "CommandAuthorization": [
      00000000,
      00000001
    ],
    "ChatAuthorization": [
      /* Users in command authorization are automatically added. */
      00000006
    ],

    "DevelopmentMode": true
  }
}
```

Figuur 24 ColimbraBot configuratie

De `Watson` sectie is hier leeg gelaten. Binnen deze sectie wordt het `WatsonCore` project geconfigureerd. Het schema van deze configuratie is beschreven in paragraaf 12.2, in deze paragraaf staat een soortgelijk voorbeeld in figuur 20.

De `TelegramBot` sectie is toegevoegd om de `ColimbraBot` klasse te configureren.

De `Token` geeft aan welke token hoort bij de `TelegramBot`. Deze token is gegenereerd door Telegram om een Telegram bot te kunnen implementeren.

In deze configuratie kunnen autorisaties toegevoegd worden om commando's uit te voeren en om met de `ColimbraBot` te mogen communiceren. Elke gebruiker die geautoriseerd is om commando's uit te voeren is automatisch geautoriseerd om met `Watson` te mogen communiceren.

De autorisatie om met de `ColimbraBot` te communiceren (in de `ChatAuthorization` sectie) wordt alleen gebruikt als de applicatie niet in ontwikkelmodus staat. Ontwikkelmodus wordt aangegeven met de `DevelopmentMode` tag.

13. Conclusie en aanbevelingen

Dit hoofdstuk bevat de uiteindelijke conclusie van de onderzoeksfase en de aanbevelingen voor het uitbreiden van het prototype. In paragraaf 13.1 wordt de hoofdvraag beantwoord. Na beantwoording van de hoofdvraag zullen de aanbevelingen worden gegeven in paragraaf 13.2.

13.1 Conclusie

Zoals in de inleiding al is aangegeven beschrijven de hoofdstukken 5 tot en met 11 de onderzoeksfase. Doordat die hoofdstukken een antwoord geven op de opgestelde deelvragen kan de hoofdvraag beantwoord worden.

Deze paragraaf geeft een antwoord op de volgende hoofdvraag:

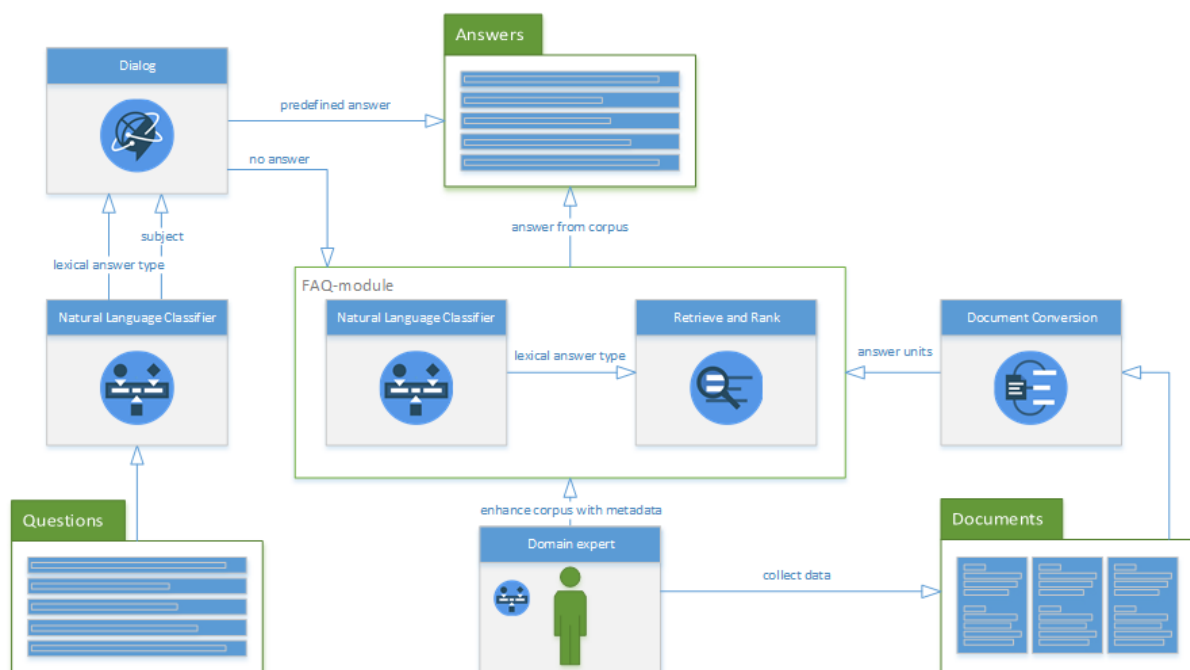
Op welke manier kunnen vragen binnen de verzekeringsbranche, in natuurlijke taal, beantwoord worden met behulp van IBM Watson en hoe kan de werking hiervan bewezen worden met een prototype (proof of concept)?

Om deze vraag te beantwoorden is gekeken naar de beschikbare Watson services en welke van deze services relevant zijn om het gewenste prototype te realiseren.

Vragen binnen de verzekeringsbranche kunnen beantwoord worden met behulp van Watson door in ieder geval gebruik te maken van de volgende Watson services:

- Dialog service;
- Document Conversion service;
- Natural Language Classifier service;
- Retrieve and Rank service.

De manier waarop een vraag- en antwoordsysteem opgesteld kan worden met deze services is door de architectuur uit figuur 25 hieronder te gebruiken. Meerdere informatiebronnen kunnen toegevoegd worden door meerdere modules te gebruiken. Bijvoorbeeld door gebruik te maken van een module met polisvoorwaardendocumenten en een module om Wikipedia door te zoeken.



Figuur 25 Architectuur - conclusie

De Document Conversion service wordt gebruikt in combinatie met de Natural Language Classfier service om een corpus op te bouwen. Deze corpus wordt ingeladen in de Retrieve and Rank service waarna deze getraind wordt om vragen te kunnen linken naar de ingelezen antwoorden. De Retrieve and Rank service wordt echter alleen gebruikt om niet vooraf gedefinieerde vragen te beantwoorden met behulp van een citaat. Vaak gestelde vragen worden verwerkt in de Dialog service. De manier waarop deze vaak gestelde vragen herkend worden is door de Natural Language Classfier service in te zetten om het verwachte antwoordtype (lexical answer type) en de context van de vraag te achterhalen.

Het prototype waarnaar verwezen wordt in de hoofdvraag zal eind juni 2016 gepresenteerd worden op een verzekeringscongres. Voor de realisatie van dit prototype is het vraag- en antwoordsysteem verwerkt in de bestaande chatapplicatie "Telegram". Telegram maakt het mogelijk om een chatbot te implementeren waarmee gecommuniceerd kan worden. Omdat Telegram een bestaande chatapplicatie is zijn verschillende interfaces beschikbaar om met het systeem te communiceren.

13.2 Aanbevelingen

Deze paragraaf zal een aantal aanbevelingen opsommen op functioneel en technisch gebied. Deze aanbevelingen geven aan op welke manier de Watson implementatie uitgebreid kan worden of in een andere omgeving ingezet kan worden.

Functioneel

Colimbra heeft een ontwikkelsuite, de Colimbra Application Solution (CAS), die intern gebruikt wordt om webapplicaties te ontwikkelen. In de toekomst zal Colimbra waarschijnlijk de Watson implementatie willen verwerken in CAS. Wanneer dit gebeurt zal de Dialog service overbodig worden omdat het aanmaken van de dialoogstructuur ook in CAS kan gebeuren.

Voor het prototype is besloten om een module te gebruiken waarmee vragen die regelmatig gesteld worden beantwoord kunnen worden. Aangezien deze module vaak gestelde vragen beantwoord is het beter om deze te verwerken in de Dialog service. En de modules alleen in te zetten voor vragen die niet vooraf zijn gedefinieerd.

Watson kan beter en met een hogere zekerheidsgraad vragen beantwoorden naarmate meer trainingsdata beschikbaar is. Hierdoor is het handig om een verzekeraar te benaderen die trainingsdata kan leveren. Denk hierbij bijvoorbeeld aan gesprekken van helpdeskmedewerkers of een interne database.

Technisch

Wanneer de Dialog service wordt vervangen door CAS zal de `WatsonInstance` klasse aangepast moeten worden. De `WatsonInstance` klasse implementeert de architectuur van het vraag- en antwoordsysteem, deze architectuur kan ook gedefinieerd worden in CAS. Hierdoor moet CAS echter uitgebreid worden om de benodigde Watson services aan te kunnen roepen.

14. Procesgang evaluatie

Dit hoofdstuk geeft een evaluatie op het proces dat verlopen is tijdens het onderzoek en het implementeren van het prototype.

Aan het begin van de afstudeerperiode waren veel onduidelijkheden over de beschikbaarheid van de Watson producten. Zoals in paragraaf 5.3 omschreven is was het Watson Engagement Advisor product een geschikte keuze geweest voor de implementatie van het gewenste prototype. Na wat rondvragen bleek dit een product te zijn dat niet meer aangeboden werd. Later werd echter de indruk gewekt dat dit product wel beschikbaar zou zijn. En uiteindelijk bleek dat het product wel aangeboden kan worden maar aangeraden wordt om de Bluemix services te gebruiken.

Een ander probleem was het opzetten van de architectuur. Een vraag- en antwoordsysteem is een techniek waarover IBM beschikt. De services van Bluemix bestaan echter uit losse modules waarmee zelf een vraag- en antwoordsysteem gerealiseerd moet worden. Dit kan natuurlijk op verschillende manier met verschillende services maar nergens was te vinden wat de beste praktijken zijn voor het maken van een vraag- en antwoordsysteem zoals dat bijvoorbeeld gebeurde met het Watson Engagement Advisor product.

Door deze onduidelijkheden is de afstudeerperiode niet geheel verlopen zoals beschreven in het plan van aanpak (bijlage A). Het onderzoek is uitgelopen en het verloop van de onderzoeksfase is gepaard gegaan met de ontwikkeling van het prototype. Hiervoor is gekozen om nu gedurende het onderzoek gelijk een bevestiging gegeven kon worden. Voor het onderzoeken van de werking van een service moest die service vaak gewoon in gebruik genomen worden.

Gedurende de afstudeerperiode is veel kennis opgedaan. Ondanks dat de machine learning aspecten grotendeels door IBM is verwerkt in de verschillende services, is veel geleerd over de techniek achter machine learning en de manier waarop gebruik gemaakt kan worden van applicatie die machine learning technieken implementeren.

Omdat gekozen is om het nieuwe .NET core framework te gebruiken is ook veel kennis opgedaan over de werking van dit framework. Dit kan handig zijn in de toekomst wanneer het .NET core framework vaker gebruikt gaat worden.

15. Bibliografie

Apache Software Foundation. (2016). *APACHE SOLR*. Opgeroepen op 06-05-2016, van apache:
<http://lucene.apache.org/solr/>

Best, J. (2015). *Tech Republic*. Opgeroepen op 15-02-2016, van IBM Watson: The inside story of how the Jeopardy-winning supercomputer was born, and what it wants to do next:
<http://www.techrepublic.com/article/ibm-watson-the-inside-story-of-how-the-jeopardy-winning-supercomputer-was-born-and-what-it-wants-to-do-next/>

Campbell, M. (2002). *Artificial Intelligence*. USA: Elsevier. Opgehaald van:
http://ac.els-cdn.com/S0004370201001291/1-s2.0-S0004370201001291-main.pdf?_tid=7b4df6d0-d61b-11e5-932a-00000aab0f6c&acdnat=1455785102_5325cb0fccff072f174de5179e2acb76

Colimbra. (2015). *Colimbra*. Opgeroepen op 12-02-2016, van Colimbra:
<http://www.colimbra.com/home/>

IBM. (2016a). *IBM Bluemix*. Opgeroepen op 10-02-2016, van IBM Bluemix:
<http://www.ibm.com/cloud-computing/bluemix/>

IBM. (2016b). *IBM Watson*. Opgeroepen op 02-02-2016, van IBM:
<http://www.ibm.com/smarterplanet/us/en/ibmwatson/index.html>

IBM. (2016c). *IBM Watson Gallery*. Opgeroepen op 02-15-2016, van IBM Watson Developer Cloud:
<http://www.ibm.com/smarterplanet/us/en/ibmwatson/developercloud/gallery.html>

IBM. (2016d). *Watson Services*. Opgeroepen op 09-02-2016, van IBM Watson Developer Cloud:
<https://www.ibm.com/smarterplanet/us/en/ibmwatson/developercloud/services-catalog.html>

Kleeff, I. v. (2015). *IBM Watson businesscase* - Colimbra. Hoofddorp.

Massé, M. (2012). *REST API Design Rulebook*. Opgeroepen op 02-02-2016, van:
https://books.google.nl/books?hl=en&lr=&id=eABpzyTcJNIC&oi=fnd&pg=PR3&dq=REST+api&ots=vyTB--maKz&sig=PD9L8rZsb9qavNszeUPrVU_8_Os#v=onepage&q=REST%20api&f=false

MongoDB. (2016). *mongoDB*. Opgeroepen op 18-05-2016, van mongoDB:
<https://www.mongodb.com/>

Ravencroft, J. (2016). *Conference call: Colimbra & Watson Technical Session*.

Sips, R. (2016). *Conference: Colimbra & Watson Technical meeting* - IBM. Amsterdam.

Telegram. (2016). *Telegram Bot API*. Opgeroepen op 15-04-2016, van Telegram:
<https://core.telegram.org/bots/api>

.Net Foundation. (2016). *.Net Core*. Opgeroepen op 05-04-2016, van GitHub:
<https://dotnet.github.io/>

16. Bijlagen

Vanuit verschillende plekken in dit document wordt verwezen naar een aantal bijlagen. Het plan van aanpak is voorafgaand aan het maken van deze scriptie opgesteld en vanuit de opleiding verplicht als eerste bijlage.

In de word versie van dit document kan op de onderstaande icoontjes geklikt worden om de losse versies van de bijlagen te openen. Voor versies op een ander medium zijn de bijlagen te vinden in de hierop volgende pagina's.

Bijlage A: Plan van Aanpak



Plan van aanpak

Bijlage B: Functioneel ontwerp



Functioneel
ontwerp

Bijlage C: Technisch ontwerp



Technisch ontwerp

Bijlage D: Watson serviceoverzicht



Watson
serviceoverzicht

Bijlage E: Prototype



Prototype

Bijlage A: Plan van Aanpak

Plan van aanpak

Colimbra – IBM Watson implementatie

Versie 1.0

Auteur

Rick Koster

S: 1622528

E: rick.koster@student.hu.nl

E: rkoster@colimbra.com

Legenda

S: Studentnummer

In opdracht van

Hogeschool Utrecht

Nijenoord 1

3552 AS Utrecht

T: 088 481 82 83

E: info@hu.nl

T: Telefoonnummer

Stagebedrijf

Colimbra B.V.

Kruisweg 821

2132 NG Hoofddorp

T: 023 565 37 38

E: info@colimbra.com

E: E-mailadres

Inhoudsopgave

1. Inleiding.....	6
2. Colimbra	8
2.1 Bedrijfscontext	8
2.2 Kwestie.....	8
2.3 Aanleiding	9
2.4 Doelstelling	9
3. International Business Machines (IBM)	10
3.1 IBM Watson	10
3.2 Geschiedenis Watson.....	10
3.3 Samenwerking.....	10
3.4 Watson producten	10
3.5 Voorbeeld applicaties	11
4. Theoretisch kader	12
4.1 Centrale begrippen	12
4.2 Modellen en methoden	13
4.3 Watson alternatieven	15
5. Opdracht	16
5.1 Omschrijving	16
5.2 Productoplevering.....	16
5.3 Onderzoeksvragen	17
5.4 Deelresultaten.....	19
5.5 Kwaliteitscriteria	20
5.6 Voorbeeld.....	21
6. Organisatie	22
6.1 Algemene werkwijze	22
6.2 Onderzoeksfase.....	23
6.3 Ontwikkelfase	24
6.4 Risico's en maatregelen	25

7. Tijdlijn.....	26
7.1 Schoolzaken	26
7.2 Onderzoeksfase.....	27
7.3 Ontwikkelfase	27
7.4 Totaalbeeld	28
8. Contactgegevens.....	29
8.1 Afstudeerder	29
8.2 Hogeschool Utrecht	29
8.3 Colimbra.....	30
8.4 International Business Machines (IBM)	30
9. Afkortingen en begrippen	31
9.1 Afkortingen	31
9.2 Begrippen.....	31
10. Bibliografie	33
11. Bijlage.....	35
11.1 Bijlage A: Interview – Ronald van den Berg	35

1. Inleiding

Dit plan van aanpak is opgesteld in opdracht van Hogeschool Utrecht als onderdeel van het afstuderen. Binnen dit document zullen de afstudeerlocatie, de afstudeeropdracht, de werkwijze en de bijbehorende opleveringsdata worden omschreven.

Colimbra

De afstudeeropdracht wordt uitgevoerd in opdracht van Colimbra. Colimbra is een IT-partner die software ontwikkelt, beheert, implementeert en onderhoudt binnen de verzekeringsbranche. Meer informatie over Colimbra, inclusief de aanleiding en doelstelling voor de uitvoering van deze opdracht, is te vinden in hoofdstuk 2.

International Business Machines (IBM)

Gedurende de afstudeeropdracht wordt veel onderzoek gedaan naar het Amerikaanse bedrijf International Business Machines (hierna IBM) aangezien de opdracht wordt uitgevoerd met behulp van IBM Watson. Watson is een supercomputer van IBM die natuurlijke taal begrijpt en getraind kan worden om een bepaalde sector/branche, in dit geval de verzekeringsbranche, te begrijpen. Uiteindelijk zal Watson in staat moeten zijn om vragen, gesteld in natuurlijke taal, te beantwoorden. Een beschrijving van IBM Watson is in hoofdstuk 3 te vinden. Om Watson toegankelijk te maken is Colimbra lid geworden van het IBM Watson ecosysteem partnerschap. Ook dit partnerschap wordt in hoofdstuk 3 beschreven.

Theoretisch kader

IBM Watson is gekozen als oplossing voor de opdracht. Watson is echter niet het enige systeem met de mogelijkheid om, in natuurlijke taal, vragen te beantwoorden. In hoofdstuk 4 wordt het theoretische kader van deze opdracht beschreven.

Opdracht

De wens is om een prototype te ontwikkelen waarmee de uiteindelijke klanten van Colimbra hun verzekeringsgegevens, via een dialoog, kunnen benaderen met behulp van IBM Watson. Dit houdt in dat eerst een onderzoek uitgevoerd moet worden naar de werking en de mogelijkheden van Watson. Hierdoor kan de afstudeerperiode worden ingedeeld in een onderzoeksfase en een ontwikkelfase. De onderzoeksfase is voorbereidend op de ontwikkelfase, in de onderzoeksfase zal worden onderzocht hoeveel werk verricht moet worden in de ontwikkelfase. Tijdens de onderzoeksfase zal onder andere een technisch ontwerp en een functioneel ontwerp worden gemaakt. Meer informatie over de opdracht is beschreven in hoofdstuk 4.

Organisatie

Zoals hiervoor al is vermeld bestaat de afstudeerperiode uit een onderzoeks- en een ontwikkelfase. De ontwikkelfase zal een wat abstracte planning aanhouden, gedurende de ontwikkelfase zal eerst een globale planning worden opgezet, deze zal daarna per week worden afgebakend in "sprints" wat een SCRUM-werkwijze suggereert. Een gedetailleerdere omschrijving van de organisatie is gegeven in hoofdstuk 6.

Tijdslijn

Om de afstudeerperiode in goede banen te laten leiden zal een planning worden gemaakt. Tijdens de onderzoeksfase wordt een wat abstracte planning gehanteerd aangezien veel aspecten (vooral over IBM Watson) dan nog onduidelijk zijn. De ontwikkelfase zal een striktere planning hebben. Meer informatie over de tijdslijn wordt gegeven in hoofdstuk 7. In dit hoofdstuk worden schoolzaken en de data omtrent de onderzoeksfase en de ontwikkelfase eerst apart omschreven. Uiteindelijk wordt een totaalbeeld gegeven wat de planning voor de gehele afstudeerperiode weergeeft.

Contactgegevens

In het verloop van dit verslag zullen de volgende termen gebruikt worden: "afstudeerder", "docentbegeleider", "bedrijfsbegeleider" en "relationship manager". Met deze termen worden de volgende personen bedoeld:

- Afstudeerder Rick Koster
- Docentbegeleider Henk van Nimwegen
- Bedrijfsbegeleider Ronald van den Berg
- Relationship manager Joshua Hall

De contactgegevens inclusief functie van deze personen en andere, voor deze afstudeerperiode, relevante personen zijn toegevoegd in hoofdstuk 8.

Afkortingen en begrippen

Voor de volledigheid is in hoofdstuk 9 een alfabetisch gesorteerde lijst te vinden van afkortingen die gebruikt zijn in dit document inclusief een volledige lijst van gebruikte begrippen. De centrale begrippen uit het theoretische kader zijn ook verwerkt in deze lijst.

Bibliografie

Een alfabetisch gesorteerde lijst van gebruikte literatuur wordt gegeven in hoofdstuk 10. Deze lijst is weergegeven in APA-formaat.

Bijlage

Aan het eind van dit document wordt, onder hoofdstuk 11, een bijlage toegevoegd. Binnen dit document wordt naar deze bijlage verwezen. Het gaat hierbij om de onderstaande bijlage:

- Bijlage A: Interview – Ronald van den Berg.

2. Colimbra

Binnen dit hoofdstuk wordt een omschrijving gegeven van Colimbra. De bedrijfscontext wordt omschreven in paragraaf 2.1. De kwestie waaraan gewerkt wordt is omschreven in paragraaf 2.2. De aanleiding wordt gegeven in paragraaf 2.3 met daarna in paragraaf 2.4 de uiteindelijke doelstelling.

2.1 Bedrijfscontext

Zoals in de inleiding al is aangegeven, zal de afstudeeropdracht uitgevoerd worden in opdracht van Colimbra. Colimbra is een IT-partner (Information Technology partner) die software ontwikkelt, beheert, implementeert en onderhoudt binnen de verzekeringsbranche. Colimbra heeft een relatief kleine omvang van ongeveer 20 werknemers waarvan de meeste domeinexperts zijn.

Uitvoering van de opdracht zal resulteren in een onderzoeksverslag. De wens is om ook een losstaand prototype te realiseren (proof of concept). Deze producten zullen gerealiseerd worden op de ontwikkelafdeling van Colimbra. Uiteindelijk kan deze service resulteren in een toevoeging aan de 'Mijn omgeving' service, een service aangeboden door Colimbra waarmee alle polis gegevens van de uiteindelijke klant op overzichtelijke wijze bij elkaar worden gehouden (Kleeff, 2015).

Colimbra richt zich op drie gebieden: front- en midoffice oplossingen als extranetten en doelgroep omgevingen, koppelvraagstukken voor berichtenverkeer en documentuitwisseling, en klantbedieningsconcepten.

2.2 Kwestie

Tijdens de stageperiode zal gewerkt worden aan het eenvoudiger maken om vragen te stellen binnen de verzekeringsbranche. Met een verzekeringsvraag is het vaak zo dat door polis documenten en de bijbehorende polisvoorwaarden gezocht moet worden. Als er dan nog onduidelijkheden zijn kan de helpdesk van het betreffende verzekeringsbedrijf worden gebeld. Dit zorgt echter voor extra kosten voor het desbetreffende verzekeringsbedrijf (klant van Colimbra) en de vragende persoon (uiteindelijke klant).

Om deze kwestie te verbeteren wil Colimbra een onderzoek uitvoeren naar de werking van IBM Watson en de hoeveelheid werk wat verricht moet worden voor de ontwikkeling van een digitale helpdesk. Watson is interessant voor deze kwestie omdat Watson, zoals eerder in de inleiding al is aangegeven, in staat is om natuurlijke taal te begrijpen en daardoor een antwoord te geven op gestelde vragen.

Dit zal het beantwoorden van vragen eenvoudiger maken voor de uiteindelijke klant omdat het niet nodig is om zelf documenten te doorzoeken. Aangezien het een digitale helpdesk betreft is het voor de uiteindelijke klant mogelijk om kosteloos vragen te stellen en kan de drempel om een vraag te stellen lager liggen. Dit omdat het niet nodig is om een helpdeskmedewerker te bellen.

2.3 Aanleiding

Binnen de verzekeringsbranche worden een groot aantal data en documenten verwerkt. Deze data en documenten kunnen zoals eerder al aangegeven met behulp van Watson worden omgezet tot kennis, waardoor via een dialoog gezocht kan worden naar deze gegevens. De aanleiding om deze opdracht uit te voeren is dan ook om het bemachtigen van verzekeringsdata of documenten eenvoudiger te maken voor de uiteindelijke klant. Door de mogelijke verlaging van vragen die naar de helpdesk gaan kan hierdoor ook worden bespaard op kosten.

De data waarover gesproken wordt is bijvoorbeeld afkomstig uit een dekkingsoverzicht of een opgestelde polis. Wanneer IBM Watson deze documenten op een juiste manier inleest en begrijpt zullen bijvoorbeeld de volgende vragen beantwoord kunnen worden:

- **Als ik met mijn auto in het buitenland tegen een paaltje aan rijd, wordt mijn schade dan vergoed door de verzekering?**
- **Heb ik een extra verzekering nodig als ik op wintersport ga?**
- **Ik ben laatst getrouwd, welk effect heeft dit op mijn verzekeringsdekking?**
- **Wat zijn de dekkingen voor sieraden?**

Volgens een vooraf opgestelde business-case door (Kleeff, 2015), is ook de bedrijfsstrategie van Colimbra een belangrijke aanleiding voor het oppakken van deze kwestie: "Colimbra wil nieuwe producten aan blijven bieden voor hun bestaande en potentieel nieuwe klanten". Colimbra wil met het oppakken van deze kwestie inspelen op de vraag naar innovatie vanuit de verzekeringsbranche. Dit geeft Colimbra de kans om vernieuwende producten aan te blijven bieden voor hun bestaande en potentieel nieuwe klanten.

2.4 Doelstelling

De situatie waar Colimbra zich nu in bevindt is dat Watson interessant klinkt door de mogelijkheden van Watson. Dat het project uitvoerbaar is staat vast. Het is echter nog onduidelijk in hoeverre de beschikbare tools/services van Watson ontwikkeld zijn en wat er dus moet gebeuren om een prototype te kunnen ontwikkelen.

Het doel van de opdracht is om voor Colimbra vast te stellen of Watson toegevoegde waarde biedt aan hun bestaande applicaties. Dit zal zo zijn als blijkt dat Watson een volledig ontwikkelde set van tools levert. Na afloop van de afstudeerperiode moet aan de hand van een onderzoeksverslag worden aangetoond hoeveel werk het ontwikkelen van een Watson helpdesk met zich meebrengt. De wens is om na afloop van de afstudeerperiode ook een prototype (proof of concept) gerealiseerd te hebben, inhoudelijk zal dit prototype afhangen van het onderzoeksverslag.

3. International Business Machines (IBM)

Dit hoofdstuk zal in paragraaf 3.1 aangeven waar het bedrijf IBM zich mee bezig houdt, met de focus op IBM Watson. Om te begrijpen wat het oorspronkelijke doel van Watson is en wat het doel is voor in de toekomst wordt in paragraaf 3.2 de geschiedenis van Watson besproken. Paragraaf 3.3 zal aangeven hoe de samenwerking zal verlopen met IBM. Daarna wordt in paragraaf 3.4 aangegeven welk Watson product het meest interessant is. Tot slot zal paragraaf 3.5 verwijzen naar een aantal voorbeeld applicaties. De informatie uit dit hoofdstuk is, tenzij anders aangegeven, afkomstig uit de officiële IBM Watson website (IBM, 2016). Dit hoofdstuk kan als onderdeel van het theoretisch kader beschouwd worden.

3.1 IBM Watson

International Business Machines (IBM) is een Amerikaans bedrijf dat zich bezighoudt met de ontwikkeling van hardware, middleware en software, en met het bieden van infrastructuur, hosting en consultancy services.

Watson is een supercomputer van IBM die de Engelse natuurlijke taal begrijpt en getraind kan worden om een bepaalde sector/branche, in dit geval de verzekeringsbranche, te begrijpen. Dit gebeurt door de mogelijkheid van Watson om documenten of webpagina's om te zetten naar kennis. Uiteindelijk kan Watson dus getraind worden als 'expert' binnen de verzekeringsbranche, wat het onder andere mogelijk maakt om Watson in te zetten als een digitale helpdesk.

3.2 Geschiedenis Watson

Het bedrijf IBM heeft een aantal momenten waarop een "Grand Challenge" wordt aangegaan. Tijdens deze uitdaging wordt ontwikkeld aan een machine die tegenover de mens wordt gezet. Eén van deze uitdagingen heeft geresulteerd in de ontwikkeling van Deep Blue. Deep Blue is de machine die de wereldkampioen schaken, Garry Kasparov, heeft verslagen in 1997 (Campbell, 2002).

Watson is net zoals Deep Blue ontwikkeld als "Grand Challenge". Deze keer is Watson gericht op het winnen van "Jeopardy!", een Amerikaanse tv-show waarin een quiz wordt gespeeld. Tijdens deze quiz worden antwoorden gegeven waarna de deelnemers de bijbehorende vraag moeten verzinnen. Om deze quiz te winnen moest Watson dus eerst het antwoord begrijpen en daar daarna de bijbehorende vraag bij formuleren. Watson heeft uiteindelijk gewonnen van de, in die tijd, twee beste spelers die hebben meegespeeld: Ken Jennings en Brad Rutter (Best, 2015).

3.3 Samenwerking

Zoals al eerder aangegeven is een partnerschap opgezet tussen Colimbra en IBM. Dit is een "shared revenues partnership" wat inhoudt dat beide bedrijven investeren in de ontwikkeling van het product en uiteindelijk de winst zullen delen. Het aangaan van dit partnerschap betekend onder andere dat er momenten zijn waarop vanuit IBM geregeld wordt dat informatie, specifiek gericht op de use-case van Colimbra, wordt verschaft. Dit partnerschap houdt ook in dat Colimbra toegang krijgt tot de producten aangeboden vanuit het Watson Ecosysteem en gericht vragen kan stellen aan Watson experts binnen IBM.

3.4 Watson producten

Het hiervoor genoemde Watson Ecosysteem biedt toegang tot een aantal Watson producten, namelijk de volgende: "Watson Engagement Advisor", "Watson Explorer", "Watson Discovery Advisor", "Watson for Oncology", "Watson for Clinical Trial Matching" en "Watson Knowledge Studio". Buiten deze producten, die alleen toegankelijk zijn voor IBM-partners, zijn er een aantal soortgelijke services beschikbaar zonder partnerschap. Deze zijn beschikbaar via IBM Bluemix.

Het product dat voor Colimbra het meest interessant is, is de Watson Engagement Advisor. De Watson Engagement Advisor is een technologie waarmee gecommuniceerd wordt met de gebruikers, luisterend naar vragen om een oplossing aan te bieden. Het is een vraag en antwoord systeem, de gebruiker kan vragen stellen in natuurlijke taal. Watson begrijpt wat de gebruiker wil doen, analyseert de context en presenteert daarna de meest relevante oplossing. Dit product is het meest geschikt voor dit project aangezien het inzetten van Watson als digitale helpdesk precies als doel heeft om vragen te beantwoorden.

Uit een conference call met (Ravencroft, 2016) is duidelijk geworden dat dit product echter niet meer wordt aangeboden, de services van Bluemix zijn en worden verbeterd om als vervanging te kunnen dienen. Hierdoor is Colimbra in een soort bèta programma terechtgekomen, waarin verschillende aspecten nog onduidelijk zijn en bepaalde services nog niet volledig gedocumenteerd zijn. Onduidelijk is bijvoorbeeld welke Bluemix services nodig zijn en hoe deze aan elkaar gekoppeld kunnen worden om een vraag en antwoord systeem, zoals de hiervoor genoemde Engagement Advisor, te kunnen realiseren.

Zoals in het theoretisch kader al was aangegeven zijn deze Watson producten/services toegankelijk met behulp van een REST-API. Dit houdt in dat elke service aangesproken kan worden met behulp van webrequests (Massé, 2012).

3.5 Voorbeeld applicaties

De onderstaande applicaties zijn opensource demo applicaties die gebruik maken van een of meerdere Watson services. Deze applicaties tonen een aantal mogelijkheden van Watson. De applicaties in deze paragraaf zijn gemaakt met behulp van de binnen Bluemix geleverde services en zijn te vinden via de IBM Watson Gallery (IBM, 2016).

- **Questions on the Natural Language Classifier**

Deze applicatie maakt het mogelijk om in natuurlijke taal vragen te stellen over de Natural Language Classifier service. De "Questions on the Natural Language Classifier" applicatie is ontwikkeld in Java en deze applicatie maakt gebruik van één Watson service, namelijk de Natural Language Classifier service.

Demo: <http://watson-on-classifier.mybluemix.net/>

- **What's in theaters?**

Deze applicatie maakt het mogelijk om, met behulp van natuurlijke taal, huidige of verwachte films te vinden aan de hand van een aantal criteria. Net zoals de vorige applicatie is ook de "What's in theaters" applicatie ontwikkeld in Java. Deze applicatie heeft toegang tot de database van themoviedb.org om filmgegevens op te halen. Verder gebruikt deze applicatie twee Watson services, de Natural Language Classifier service en de Dialog service. De werking van deze applicatie is gedocumenteerd waardoor het in combinatie met de opensource code een goed voorbeeld is voor een casestudy.

Demo: <http://watson-movieapp-nlcdialog.mybluemix.net/watson-movieapp-dialog/dist/index.html#/>

- **News explorer**

Deze applicatie maakt automatisch een nieuws informatienetwerk en toont grote hoeveelheden nieuwsresultaten op een begrijpelijke manier. De "News explorer" applicatie is ontwikkeld in Java en maakt gebruik van de AlchemyAPI. De AlchemyAPI is een onderdeel van Watson wat een set aan services levert. Deze set is vooral gefocust op het ophalen van informatie uit, op internet beschikbare, nieuwsartikelen en blogs.

Demo: <http://news-explorer.mybluemix.net/>

4. Theoretisch kader

Dit hoofdstuk zal het theoretische kader van de afstudeeropdracht omschrijven. In paragraaf 4.1 wordt aangegeven welke begrippen centraal staan binnen dit document. Paragraaf 4.2 geeft een omschrijving van de gebruikte modellen en methoden. Uiteindelijk worden een aantal mogelijke alternatieven voor IBM Watson gegeven in paragraaf 4.3.

4.1 Centrale begrippen

Deze paragraaf zal een aantal begrippen omschrijven die centraal staan binnen dit document. Een volledige lijst van begrippen en afkortingen wordt gegeven in hoofdstuk 9.

IBM Watson

Watson is een supercomputer ontwikkeld door International Business Machines (IBM). Deze supercomputer is in staat om antwoorden te geven op vragen die gesteld zijn in natuurlijke taal. Een uitgebreidere omschrijving van IBM Watson is gegeven in hoofdstuk 3.

Machine learning

Eerder is al aangegeven dat gewerkt wordt met behulp van IBM Watson. Watson maakt veel gebruik van machine learning technieken om zijn kennis te verbeteren. Dit zijn technieken waarmee Watson, zoals de naam al doet vermoeden, in staat is om te leren van interacties met gebruikers en zo zijn reactie te verbeteren (IBM, 2016).

Ground truth

In het vorige begrip is aangegeven dat Watson gebruik maakt van machine learning technieken. Buiten het verbeteren van kennis is het nodig om Watson basiskennis te geven. Dit gebeurt met de Watson services meestal aan de hand van een file die wordt ingelezen en gebruikt door Watson. Deze file met trainingsmateriaal wordt vaak de "ground truth"-file genoemd (IBM, 2016).

Natuurlijke taal

Talen zoals Nederlands, Frans of Engels. De taal die in een menselijke gemeenschap van oudsher wordt gebruikt voor algemene communicatie en die een kind, opgroeiend in zo'n gemeenschap, als zijn moedertaal leert (Poecke, 1991).

IBM Bluemix

Een cloud platform als service (PaaS – platform as a service) ontwikkeld door IBM. Applicaties kunnen gehost en beheerd worden met behulp van Bluemix. Bluemix maakt het ook eenvoudiger om open source software te integreren (IBM, 2016), meer informatie over Bluemix wordt gegeven in hoofdstuk 3.

REST-API

Om Watson beschikbaar te stellen voor geïnteresseerde bedrijven zijn de services van Watson beschikbaar via een REST-API (Representational state transfer application programming interface). Dit houdt in dat de services van Watson te benaderen zijn via webrequests (Massé, 2012).

Verzekeringpolis & polisvoorwaarden

Een persoonlijk document vanuit de verzekeringsbranche waarin wordt aangegeven welke afspraken zijn gemaakt met de verzekeringsmaatschappij. De bijbehorende polisvoorwaarden bestaat uit een algemeen document vanuit de verzekeringsbranche, hierin worden de voorwaarden aan het polis document vastgelegd.

4.2 Modellen en methoden

Binnen deze paragraaf zullen een aantal modellen en methoden besproken worden die gebruikt gaan worden tijdens de afstudeerperiode. Het gaat hierbij om modellen en methoden die de loop van de afstudeerperiode begeleiden (planning) en modellen/methoden die nodig zijn om een onderzoeksverslag, een functioneel ontwerp en een technisch ontwerp op te stellen. Meer informatie over deze onderdelen worden in de loop van dit document gegeven.

Planning

Om de ontwikkelfase in goede banen te leiden zal gewerkt worden met een beknopte SCRUM-werkwijze. De onderstaande informatie over SCRUM is afkomstig uit (Overheem, 2004). De volgende onderdelen van SCRUM worden overgenomen tijdens de ontwikkelfase:

- **Sprints**
Met een SCRUM-werkwijze worden een aantal sprints georganiseerd van 2-4 weken. Dit project zal echter wekelijkse sprints hanteren aangezien het in dit geval een (klein) eenmansproject betreft.
- **Back log**
Een back log wordt uitgewerkt om overzichtelijk aan te tonen welke functionaliteit is geïmplementeerd in een vorige sprint/iteratie.

De overige onderdelen van SCRUM worden niet overgenomen omdat dit voor een onnodig grote hoeveelheid overhead zorgt, vooral aangezien de afstudeeropdracht een eenmansproject betreft.

Onderzoeksmethoden

Tijdens de uitvoering van het onderzoek worden een aantal vragen beantwoord aan de hand van de onderstaande onderzoeksmethoden:

- **Interview**
Antwoorden op een opgestelde vragenlijst. Wanneer deze onderzoeksmethode gebruikt wordt zal de vragenlijst inclusief antwoorden altijd worden toegevoegd aan de bijlage van het onderzoeksverslag/scriptie.
- **Groepsgesprek**
Er zal een aantal keren een conference call worden gehouden met IBM. Relevante informatie uit de conference calls wordt genotuleerd voor referentie.
- **Literatuuronderzoek**
Een groot aantal informatie is beschikbaar via internet. Deze informatie wordt alleen gebruikt als het gaat om wetenschappelijke artikelen, officiële documentatie of nieuwsartikelen waarin gerefereerd wordt zoals tijdens dit onderzoek.
- **Observatie**
Deze onderzoeksmethode wordt gebruikt wanneer nieuwe informatie kan worden achterhaald uit beschikbare informatie. Bijvoorbeeld uit voorgaande deelvragen.
- **Casestudy**
Een casestudy houdt in dat gekeken wordt naar het antwoord/de oplossing van een soortgelijke vraag/probleem. Deze oplossing wordt bestudeerd om te begrijpen welke oplossing gebruikt kan worden voor de te beantwoorde vraag.

Functioneel ontwerp

Zoals in de inleiding al werd vermeld wordt een functioneel ontwerp gerealiseerd. Dit functioneel ontwerp zal in ieder geval de onderstaande methode/modellen bevatten:

- **MoSCoW methode**

De MoSCoW methode omschrijft de "must", "should", "could" en "won't" have onderdelen van een applicatie. Dit houdt in dat wordt omschreven welke onderdelen de applicatie moet hebben om te kunnen functioneren, welke onderdelen de applicatie zou moeten hebben, welke onderdelen eventueel toegevoegd kunnen worden en uiteindelijk welke onderdelen niet in de applicatie terecht komen (Stapleton, 2003). Volgens (Apello, 2008) zou de MoSCoW methode geen effect hebben omdat klanten alles zien als een must. Deze MoSCoW methode wordt echter opgesteld in overleg met de bedrijfsbegeleider die een technische achtergrond heeft en begrijpt welke onderdelen meer moeite kosten om te implementeren.

- **Use-case diagram**

Een use-case diagram is bedoeld om met behulp van een diagram duidelijkheid te geven over het gebruik van een applicatie. Hierin is te zien welke gebruikers er zijn en hoe deze gebruikers gebruik maken van de applicatie (Maßen & Lichter, 2002).

Technische ontwerp

Een technisch ontwerp wordt gemaakt om duidelijkheid te geven over de technische aspecten van de te realiseren applicatie. In het technisch ontwerp worden de onderstaande modellen/methoden uitgewerkt:

- **Klassendiagram**

Een klassendiagram wordt opgesteld om een overzichtelijke weergave te geven van de verschillende in code opgenomen klassen achter een applicatie. Hierin is ook te zien hoe de klassen met elkaar in verband staan (Ambler, 2004).

- **Sequentiediagram**

Aangezien de applicatie gebruik maakt van een of meerdere Watson services is het handig om een sequentiediagram uit te werken. Hierin wordt getoond hoe de verschillende services met de applicatie communiceren (Ambler, 2004).

- **State Transition Diagram (STD)**

Een State Transition Diagram wordt uitgewerkt om aan te geven in welke state een applicatie zich bevindt en welke andere states beschikbaar zijn vanuit de huidige state (Ambler, 2004).

4.3 Watson alternatieven

Eerder is al aangegeven dat Watson wordt gebruikt als service voor de digitale helpdesk. Er zijn echter een aantal alternatieven. De bekendste alternatieven zijn de onderstaande:

- **Google Now**
- **Microsoft Cortana**
- **Wolfram Alpha**

De reden waarom voor IBM Watson is gekozen over de bovenstaande alternatieven is omdat IBM, zoals in hoofdstuk 3 is beschreven, een "shared revenues" partnerschap aanbiedt.

Google Now

De AI ontwikkeld door Google. Google Now is vooral bekend door het Android operating system voor mobiele telefoons, waarin het functioneert als digitale persoonlijke assistent.

Microsoft Project Oxford

Project Oxford bestaat uit een set van services aangeboden door Microsoft. Deze services komen overeen met de services die aangeboden worden vanuit IBM.

Wolfram Alpha

Een systeem gefocust op het beantwoorden van wiskundige formules. Wolfram Alpha kan wiskundige formules begrijpen in verschillende notaties en zal daarna de desbetreffende uitwerking tonen.

4.4 Voorgaande onderzoeken

IBM Watson is, zoals eerder vermeld, gekozen als oplossing voor het vraag en antwoord systeem. Vanuit IBM zijn een aantal voorbeeldapplicaties beschikbaar waarmee een vraag en antwoord systeem ontwikkeld kan worden (zie paragraaf 3.5). Dit zijn echter geen uitgebreide voorbeelden omdat uitgebreide applicaties voorheen gebruik maakte van het, niet meer beschikbare, "Watson Engagement Advisor" product. Er zijn geen uitgebreide onderzoeken gevonden waarmee de services van Bluemix aan elkaar geknoopt worden.

5. Opdracht

In dit hoofdstuk zal de afstudeeropdracht worden omschreven. Paragraaf 5.1 zal een globale omschrijving van de uit te voeren opdrachten geven. Paragraaf 5.2 geeft aan welke producten uiteindelijk opgeleverd dienen te worden met de bijbehorende kwaliteitscriteria. Paragraaf 5.3 geeft de onderzoeksvragen inclusief de vervullende functie. Daarna worden in paragraaf 5.4 de deelresultaten gegeven. In paragraaf 5.5 worden de kwaliteitscriteria gedefinieerd. Uiteindelijk zal paragraaf 5.6 met behulp van twee voorbeeldfiguren weergeven hoe de gewenste applicatie ongeveer moet functioneren.

5.1 Omschrijving

Zoals in de paragrafen hiervoor al vermeld is bestaat de opdracht uit het ontwikkelen van een prototype waarmee de uiteindelijke klanten van Colimbra, met behulp van een dialoog, verzekeringsvragen kunnen stellen. In de inleiding is al aangegeven dat deze opdracht bestaat uit een onderzoeksfase en een ontwikkelfase. De onderzoeksfase zal als doel hebben het verduidelijken van de werking van IBM Watson. Nadat deze fase is afgerond zal duidelijk zijn op welke wijze Watson ingezet kan worden en zal aan de hand van dit onderzoek een prototype worden ontwikkeld.

5.2 Productoplevering

Uiteindelijk zullen voor Colimbra twee producten worden opgeleverd. De onderzoeksfase zal uitmonden in een onderzoeksverslag (descriptief onderzoek) en de ontwikkelfase zal uiteindelijk een prototype opleveren (proof of concept).

Onderzoeksverslag

Het onderzoeksverslag moet, zoals eerder al is aangegeven, voor Colimbra duidelijk maken wat er moet gebeuren om een prototype te kunnen ontwikkelen. Hierin zal in eerste instantie per beschikbare service van Watson worden gekeken naar het doel en de werking van die service. Hierna moet duidelijk zijn welke services van belang zijn voor de realisatie van het prototype. Tijdens dit onderzoek zal de onderstaande hoofdvraag centraal staan, deze hoofdvraag is verder afgebakend tot deelvragen in de volgende paragraaf.

Op welke manier kunnen vragen binnen de verzekeringsbranche, in natuurlijke taal, beantwoord worden met behulp van IBM Watson en hoe kan de werking hiervan bewezen worden met een proof of concept (prototype)?

Deze fase wordt gezien als geslaagd wanneer de hoofdvraag beantwoord kan worden met behulp van antwoorden op de deelvragen, gegeven in de volgende paragraaf.

Prototype

Het prototype dat uitgewerkt wordt zal inhoudelijk afhangen van het voorafgaande onderzoek. De wens is dat het prototype in ieder geval een aantal, in natuurlijke taal, gestelde vragen kan beantwoorden met een, in natuurlijke taal, geformuleerd antwoord. Het antwoord zal geformuleerd worden met behulp van vooraf toegevoegde verzekeringsdocumenten. Bij toevoeging van nieuwe/aangepaste documenten zal zonder verandering aan IBM Watson de nieuwe data beschikbaar moeten zijn. Dit prototype zal dus moeten aantonen dat met behulp van Watson gegevens opgehaald kunnen worden om het handmatig zoeken door verschillende documenten te voorkomen.

Deze fase wordt gezien als geslaagd wanneer het prototype in staat is om een aantal vooraf omschreven verzekeringsvragen te beantwoorden met behulp van informatie uit een verzekeringsdocument, begeleid via een dialoog.

5.3 Onderzoeksvragen

Het onderzoek dat tijdens de onderzoeksfase uitgevoerd wordt is voorbereidend op de ontwikkelfase. Hierdoor zijn de vragen die beantwoord worden niet alleen relevant voor het onderzoeksverslag maar ook voor de scriptie.

In de vorige paragraaf is aangegeven dat de volgende hoofdvraag centraal staat in de ontwikkelfase:

Op welke manier kunnen vragen binnen de verzekeringsbranche, in natuurlijke taal, beantwoord worden met behulp van IBM Watson en hoe kan de werking hiervan bewezen worden met een proof of concept (prototype)?

Om de hoofdvraag te kunnen beantwoorden is deze afgebakend in verschillende deelvragen. Deze deelvragen zijn hieronder opgesomd. Per deelvraag wordt aangegeven wat het doel van de desbetreffende deelvraag is, wat de onderzoeksfunctie is en met behulp van welke onderzoeksmethode de vraag wordt beantwoord. In het volgende hoofdstuk zullen de onderzoeksmethoden worden beschreven. Deze deelvragen zijn opgesteld met behulp van de informatie uit hoofdstuk 3 en het theoretische kader (hoofdstuk 4).

1. Wat is IBM Watson?

Om de hoofdvraag te kunnen beantwoorden zal eerst beschreven worden wat Watson inhoudt en wat het doel is van Watson. Een antwoord op deze vraag wordt geformuleerd door te kijken naar beschikbare informatie op de IBM Watson website. Ook zal informatie beschikbaar zijn via conference calls met IBM.

Onderzoeksfunctie: Beschrijven (descriptief)

Onderzoeksmethoden: Literatuuronderzoek en groepsgesprek

2. Welke functionele eigenschappen heeft het proof of concept nodig?

Om te achterhalen welke Watson services nodig zijn zal eerst gedefinieerd worden welke functionele eigenschappen nodig zijn. Hierin zal ook worden gedefinieerd welke gegevens, genoemd in de hoofdvraag, uiteindelijk met behulp van Watson opgevraagd kunnen worden. Deze vraag zal net zoals de vorige vraag beantwoord worden met behulp van een interview met de bedrijfsbegeleider. Na dit interview zal een functioneel ontwerp worden opgesteld welke als bijlage wordt toegevoegd aan het onderzoeksverslag.

Onderzoeksfuncties: Definiëren (descriptief) en ontwerpen (prescriptief)

Onderzoeksmethode: Interview met bedrijfsbegeleider

3. Welke services zijn beschikbaar gesteld door IBM Watson, en welke van deze services zijn relevant voor dit project?

Een korte beschrijving van elke service is nodig om te achterhalen welke Watson services daadwerkelijk nodig zijn. Deze vraag zal beantwoord worden met behulp van de beschikbare servicedocumentatie. De documentatie wordt naast het functioneel ontwerp gehouden om te kunnen achterhalen welke services daadwerkelijk nodig zijn voor het prototype.

Onderzoeksfunctie: Beschrijven (descriptief)

Onderzoeksmethoden: Literatuuronderzoek en observatie

4. Hoe kunnen de relevante services van IBM Watson benaderd worden en welke configuratie mogelijkheden hebben deze services?

Wanneer de vorige vragen zijn beantwoord is duidelijk welke services uiteindelijk gebruikt moeten worden voor de ontwikkeling van een proof of concept. Voor deze services wordt beschreven wat de werking van die services is, hoe ze benaderd kunnen worden en in hoeverre elke service geconfigureerd kan worden voor een specifiek scenario. Ook deze vraag wordt beantwoord met behulp van de hiervoor genoemde servicedocumentatie.

Onderzoeksfunctie: Beschrijven (descriptief)

Onderzoeksmethode: Literatuuronderzoek

5. Hoe kunnen de relevante services aan elkaar geknoopt worden om een functionerend proof of concept te realiseren?

Om een volledig functionerende applicatie te realiseren moeten de verschillende services met elkaar samenwerken. Daarom zal een ontwerp gemaakt moeten worden waarin wordt beschreven welke gegevens van welke services beschikbaar worden gesteld voor een andere service en/of gebruiker. Tijdens het beantwoorden van deze vraag zal een technisch ontwerp worden gemaakt, deze wordt toegevoegd aan de bijlage van het onderzoeksverslag. Deze vraag zal beantwoord worden met behulp van de beschikbare servicedocumentatie. Aangezien de documentatie niet ingaat op het gebruik van meerdere services zal een voorbeeld applicatie worden bestudeerd (casestudy), en informatie worden verschaft tijdens conference calls met IBM.

Onderzoeksfuncties: Beschrijven (descriptief) en ontwerpen (prescriptief)

Onderzoeksmethoden: Literatuuronderzoek, casestudy en groepsgesprek

6. Welke extra functionaliteit, niet ondersteund door IBM Watson, moet ontwikkeld worden en hoe wordt dit gedaan?

Het is mogelijk dat niet alle benodigde functionaliteit geleverd wordt met de services van Watson. Daarom zal beschreven moeten worden welke functionaliteit mist en dus zelf gerealiseerd moet worden. Tijdens het beantwoorden van deze vraag wordt het technische ontwerp aangevuld. Om deze vraag te beantwoorden wordt gekeken naar de benodigde functionaliteit en de beschikbare functionaliteit vanuit Watson, dit is vooraf beantwoord met de deelvragen 2 en 3.

Onderzoeksfuncties: Beschrijven (descriptief) en ontwerpen (prescriptief)

Onderzoeksmethode: Observatie

7. In hoeverre dient IBM Watson getraind te worden om kennis op te doen over de verzekeringsbranche?

Watson dient getraind te worden binnen een specifieke sector voordat het vragen kan beantwoorden over die sector. In dit geval zal Watson getraind moeten worden om de verzekeringsbranche te begrijpen. Het doel van deze deelvraag is om te beschrijven hoeveel trainingsdata beschikbaar moet zijn voor de ontwikkeling van het prototype en welke inhoud deze trainingsdata moet bevatten. Deze vraag zal beantwoord worden met behulp van de beschikbare servicedocumentatie.

Onderzoeksfunctie: Beschrijven (descriptief)

Onderzoeksmethode: Literatuuronderzoek

5.4 Deelresultaten

Om de opdrachtuitvoering in goede banen te laten lopen zijn er een aantal deelresultaten die opgeleverd worden. Deze deelresultaten worden opgeleverd met als doel het aantonen van voortgang aan de docent- en de bedrijfsbegeleider en eventueel verschaffen van feedback. Hieronder worden de deelresultaten voor school, de onderzoeksfase en de ontwikkelfase gegeven. In hoofdstuk 7 kan achterhaald worden wanneer deze versies opgeleverd worden.

School

Voordat de definitieve versie van het plan van aanpak wordt ingeleverd, zullen twee conceptversies worden opgeleverd. Deze versies zijn gericht aan de docentbegeleider en de bedrijfsbegeleider.

- De eerste conceptversie geeft een beeld van de structuur van het document, deze versie zal op z'n minst de volgende onderdelen bevatten:
 - Het skelet van het document (hoofdstukindeling);
 - Een uitgewerkte inleiding;
 - Een inleidend deel bij elk hoofdstuk;
 - De punten op blz. 17 van de afstudeerleideraad (Hogeschool Utrecht, 2015).
- De tweede conceptversie zal inhoudelijk een volledig uitgewerkte versie zijn. Dit wordt echter nog steeds een conceptversie genoemd om feedback van de bedrijfs- en docentbegeleider te kunnen verwerken.

Voorafgaand aan de oplevering van de scriptie zullen, net zoals bij het plan van aanpak, twee conceptversies worden opgeleverd. Deze conceptversies zijn gericht aan de docentbegeleider.

- De eerste conceptversie van de scriptie geeft een beeld van de structuur van het document, deze versie zal in ieder geval de volgende onderdelen bevatten:
 - Het skelet van het document (hoofdstukindeling);
 - Een uitgewerkte inleiding;
 - Een inleidend deel bij elk hoofdstuk;
 - De punten op blz. 21 van de afstudeerleideraad (Hogeschool Utrecht, 2015).
- De tweede conceptversie zal de onderzoeksfase volledig verwerkt hebben in de scriptie. Dit betreft de deelvragen die tijdens de onderzoeksfase zijn beantwoord en de uiteindelijke conclusie van het onderzoek.

Onderzoeksfase

De onderzoeksfase zal zoals eerder aangegeven uiteindelijk een onderzoeksverslag opleveren. Deze fase heeft een tijdsbestek van twee maanden. Aan het eind van de eerste maand zal een conceptversie worden opgeleverd. Deze conceptversie is gericht aan de bedrijfsbegeleider.

- De conceptversie zal inhoudelijk de volgende onderdelen bevatten:
 - Het skelet van het document (hoofdstukindeling);
 - Uitgewerkte inleiding;
 - Een antwoord op de deelvragen: 1 en 2;
 - Deels/volledig antwoord op deelvraag 3.

Ontwikkelfase

Uitvoering van de ontwikkelfase levert uiteindelijk een proof of concept op. De ontwikkelfase heeft een tijdsbestek van drie maanden waarvan de laatste maand is bedoeld voor uitloop. Aan het eind van de eerste twee maanden wordt een functionerende versie opgeleverd. Deze versies zullen niet beschikken over de volledige gewenste functionaliteit maar zullen in ieder geval de ontwikkelvoortgang moeten aantonen. Deze versies zijn beide gericht aan de bedrijfsbegeleider.

- Dit prototype heeft zoals hiervoor vermeld een aantal functionele eigenschappen. De aanwezige functionaliteit is afhankelijk van de voorafgaande sprints.

5.5 Kwaliteitscriteria

De opleveringsproducten moeten aan een aantal kwaliteitscriteria voldoen, deze eisen staan hieronder vermeld. Deze kwaliteitscriteria zijn afkomstig uit een interview met de bedrijfsbegeleider (Berg, 2016). Notities over dit interview zijn toegevoegd als bijlage B in paragraaf 11.1.

Onderzoeksverslag

Het onderzoeksverslag heeft als doel om voor Colimbra duidelijk te maken hoeveel werk verricht moet worden voor de implementatie van een digitale helpdesk. Belangrijk is dat dit voldoende afgebakend wordt. Om deze eis te toetsen zijn de hoofd- en deelvragen voorgelegd aan de bedrijfsbegeleider. Ook zal vooraf een conceptversie worden opgeleverd.

Prototype

Het prototype zal uiteindelijk aan een aantal kwaliteitscriteria moeten voldoen. De kwaliteitscriteria waar het prototype aan moet voldoen zijn afkomstig uit de Must have en de Should have punten uit de MoSCoW methode. De MoSCoW methode is hieronder ingevuld naar de huidige verwachting maar kan gedurende de onderzoeksfase aangepast worden aan de hand van het onderzoek. De onderstaande tabel representeert de huidige MoSCoW methode.

MoSCoW		Watson service
Must	Dialogoog om de gebruiker te begeleiden	Dialog
	Mogelijkheid om polis documenten en polisvoorwaarden te uploaden	Document Conversion
	Informatie uit beschikbaar gestelde documenten moet te benaderen zijn via de dialogoog.	Retrieve & Rank
Should	Beter begrip van wat de gebruiker wil bereiken (de dialogoog service let alleen op zinsopbouw met behulp van keywords. Met deze toevoeging zal de intentie van de input worden begrepen waardoor beter gereageerd kan worden).	Natural Language Classifier
Could	Gesproken aansturing	Speech to Text en Text to Speech
	Admin interface om Watson te trainen	-
Won't	Ondersteuning voor de Nederlandse taal (op dit moment niet ondersteund door Watson).	
	Socialmedia analyse om informatie over de gebruiker te verzamelen	AlchemyAPI en/of Personality Insights
	Object herkenning om geüploade plaatjes te begrijpen	Visual recognition

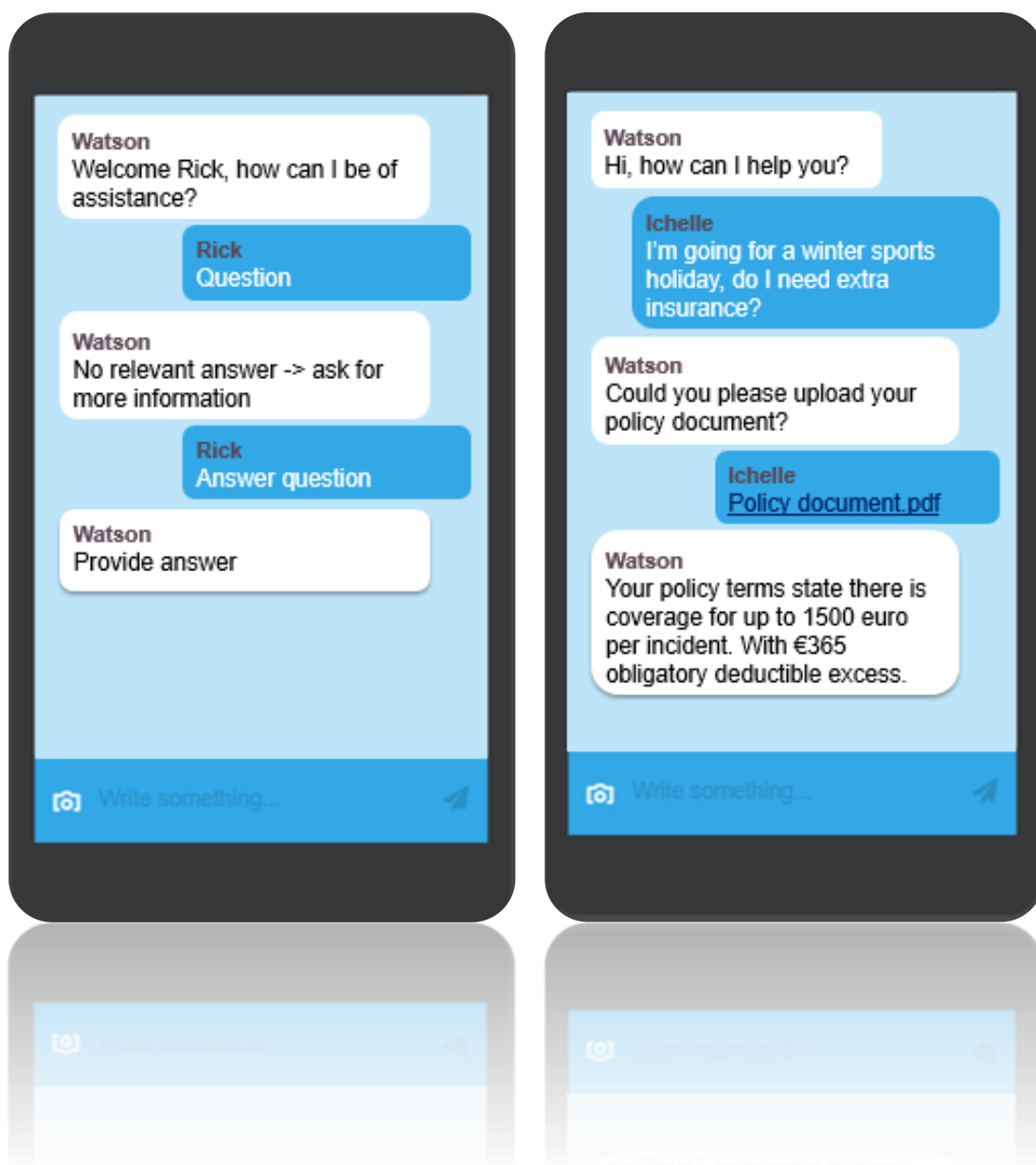
De punten in dit model worden verdeeld over wekelijkse sprints om de kwaliteitscriteria te kunnen toetsen. De werking van de MoSCoW methode en de sprints worden besproken in het volgende hoofdstuk.

5.6 Voorbeeld

Om een duidelijker beeld van de wens te geven zijn de onderstaande figuren toegevoegd. De onderstaande voorbeelden zijn concepten die alleen bedoeld zijn om een beeld van de functionaliteit te vormen. Er is geen beslissing genomen over de interface die gebruikt wordt voor deze applicatie en dit zal, aangezien het een functioneel proof of concept betreft, niet worden ontworpen binnen dit afstudeerproject.

Het linker figuur geeft een conversatieverloop weer. In dit geval heeft Watson niet genoeg informatie dus wordt een vraag teruggesteld waarna een antwoord gegeven kan worden op de initiële vraag.

Het rechter figuur laat een mogelijke conversatie zien waarin een vraag wordt gesteld over de dekking tijdens wintersport. In dit geval heeft Watson nog geen polis document van de gebruiker dus vraagt Watson of de gebruiker haar polis document kan uploaden. Dit rechter voorbeeld is overigens overgenomen uit de business case, opgesteld door (Kleeff, 2015).



6. Organisatie

Dit hoofdstuk zal omschrijven welke werkwijze wordt gehanteerd tijdens de uitvoering van deze afstudeeropdracht. Zoals in de voorgaande hoofdstukken al is aangegeven wordt de afstudeerperiode verdeeld in twee verschillende fasen, namelijk de onderzoeksfase en de ontwikkelfase. Paragraaf 6.1 zal de algemene werkwijze bespreken, in paragraaf 6.2 wordt de gehanteerde werkwijze tijdens de onderzoeksfase besproken en paragraaf 6.3 zal de werkwijze tijdens de ontwikkelfase bespreken. Tot slot zal paragraaf 6.4 de eventuele risico's en maatregelen opsommen. De werking van de methoden en modellen die in dit hoofdstuk opgenoemd worden zijn eerder omschreven in het vorige hoofdstuk.

6.1 Algemene werkwijze

De gehanteerde werkwijze tijdens de onderzoeksfase zal anders zijn dan de werkwijze tijdens de ontwikkelfase. Dit komt omdat de ontwikkelfase per deelvraag een tijds marge heeft gekregen. Contact met Hogeschool Utrecht en IBM zal echter gedurende de hele afstudeerperiode op dezelfde wijze worden georganiseerd.

Contact

Tijdens de duur van de afstudeerperiode zullen een aantal momenten voorkomen waarop contact wordt opgenomen met Hogeschool Utrecht. Dit contact zal voornamelijk met behulp van e-mails verlopen via de docentbegeleider. Zogenaamde leerteamsessies zullen georganiseerd worden via e-mail waarna een halve dag wordt vrijgemaakt om deze sessie te houden in Utrecht.

Communicatie met IBM zal in eerste instantie verlopen via e-mails met de relationship manager. Wanneer direct contact gewenst is zal dit gebeuren met behulp van een conference call of een Google hangouts call. Ook hiervoor zijn geen vaste momenten ingepland, contact wordt opgenomen wanneer er vragen zijn over Watson. In het begin van de onderzoeksfase zal, vanuit IBM, contact opgenomen worden om een aantal informatiemomenten in te plannen en het partnerschap te starten. Voor technische vragen kan onder andere gebruik worden gemaakt van een door IBM gehost en beheerd forum.

6.2 Onderzoeksfase

De ontwikkelfase zal zoals vooraf al aangegeven een wat abstracte planning aanhouden. Er wordt per deelvraag besloten hoeveel tijd beschikbaar wordt gesteld voor het beantwoorden van die deelvraag.

Gebruikte modellen

Tijdens de onderzoeksfase zal een functioneel en een technisch ontwerp worden gemaakt. Voor deze ontwerpen is hieronder aangegeven welke modellen worden uitgewerkt.

Functioneel ontwerp:

- Use-case diagram
- MoSCoW methode

Technisch ontwerp:

- Klassendiagram
- Sequentiediagram
- STD (State Transition Diagram)

In het functioneel ontwerp wordt een use-case diagram gemaakt om aan te tonen wat de huidige situatie is tegenover de gewenste situatie. De MoSCoW methode wordt uitgewerkt om te achterhalen welke functionele eigenschappen uiteindelijk ontwikkeld moeten worden.

In het technisch ontwerp is voor de genoemde modellen gekozen omdat deze modellen samen een duidelijk beeld zullen geven over de technische structuur van de applicatie. Het klassendiagram geeft voor de afstudeerder en voor eventueel toekomstige ontwikkelaars weer welke klassenstructuur is aangehouden. Het sequentiediagram geeft het verloop van de communicatie tussen de applicatie en de verschillende Watson services weer. Het STD heeft als doel om aan te geven waar de applicatie op welk moment mee bezig is.

Communicatie

Aangezien het uitvoeren van een onderzoek geen wekelijkse resultaten zal opleveren zijn er geen vaste contactmomenten ingepland met de bedrijfsbegeleider. Wanneer contact wenselijk is dan zal dit persoonlijk of via e-mail worden geregeld met de bedrijfsbegeleider.

Projectgrenzen

Om de onderzoeksfase goed te laten verlopen zijn een aantal grenzen opgesteld. Tijdens dit onderzoek wordt niet ingegaan op de volgende punten:

- Er wordt geen onderzoek gedaan naar de interne werking van Watson.
 - De manier waarop natuurlijke taal wordt ontleden;
 - De manier waarop Watson leert. Het gaat hierbij om de gebruikte machine learning technieken, wel wordt onderzocht hoe Watson ingesteld/geprogrammeerd kan worden om te leren van interacties met gebruikers.

Kwaliteitsbewaking

Zoals eerder al is aangegeven zal een conceptversie van het onderzoeksverslag worden opgeleverd. Deze oplevering zorgt ervoor dat de bedrijfsbegeleider kan verifiëren dat het verslag de juiste richting op gaat.

6.3 Ontwikkelfase

De ontwikkelfase zal worden gestart met het maken van een globale planning. Dit gebeurt pas aan het begin van de ontwikkelfase omdat na afronding van de onderzoeksfase duidelijk zal zijn wat voor werk verricht moet worden voor de ontwikkeling van een prototype. Na het opstellen van deze globale planning zal wekelijks worden gekeken naar de geïmplementeerde functionaliteit uit de voorgaande sprint en wordt bepaald welke functionaliteit in de volgende sprint wordt uitgewerkt.

Gebruikte modellen

Eerder is al aangegeven dat de ontwikkelfase wordt doorlopen met behulp van SCRUM. De volgende onderdelen zijn overgenomen uit de SCRUM-ontwikkelmethode:

- Sprints;
- Back log.

De SCRUM-ontwikkelmethode is gekozen omdat met de aanpassingen besproken in het theoretische kader (paragraaf 4.2) de wekelijkse sprints een goed beeld geven over de voortgang van de ontwikkeling. Hierdoor kan de bedrijfsbegeleider indien nodig feedback geven over ontwikkelde onderdelen.

Communicatie

Tijdens de ontwikkelfase zal wekelijks een communicatiemoment met de bedrijfsbegeleider worden gepland. Dit moment zal aan het begin of aan het eind van de week worden gehouden met als doel om de voortgang te bespreken en te bepalen welke punten worden uitgewerkt in de volgende sprint. De afstudeerder zal elke keer met een voorstel komen wat besproken wordt tijdens dit communicatiemoment.

Projectgrenzen

Voor de ontwikkelfase zijn net zoals bij de onderzoeksfase een aantal grenzen opgesteld. De onderstaande punten worden niet uitgewerkt in de ontwikkelfase:

- Er wordt geen aandacht besteed aan het ontwerpen van een userinterface. Dit project betreft een proof of concept om de werking van Watson aan te tonen, een klantvriendelijke interface zal in een ander stadium worden ontwikkeld. Wel zal een simpele interface, mogelijk zonder uitgebreide styling, worden gemaakt om gedurende de ontwikkeling met Watson te kunnen communiceren.
- Er wordt geen ondersteuning voor de Nederlandse taal geïmplementeerd. Het proof of concept wordt in het Engels getraind en benaderd aangezien Watson de Engelse natuurlijke taal begrijpt. Voor ondersteuning van de Nederlandse taal moet gewacht worden op ondersteuning vanuit IBM, of de input moet eerst van Nederlands naar Engels vertaald worden waarna de output weer van Engels naar Nederlands vertaald moet worden.

Kwaliteitsbewaking

Om de kwaliteit van het prototype te toetsen zijn zoals eerder aangegeven een aantal demoversies die opgeleverd worden. Deze versies zorgen net zoals bij het onderzoeksverslag voor verificatie.

6.4 Risico's en maatregelen

Hieronder wordt opgesomd welke risico's de uitvoering van dit project met zich meebrengt. Per risico wordt aangegeven welke werkwijze gehanteerd wordt om het risico te voorkomen (preventie) en wat er gebeurt wanneer het beschreven risico toch optreedt (reactie).

- **Het implementeren van Watson blijkt te veel werk te worden, de services die Watson beschikbaar stelt zijn niet voldoende om een functionerend prototype te kunnen ontwikkelen binnen het tijdsbestek van deze afstudeerperiode.**

Preventie Dit risico is helaas niet te vermeiden aangezien dit risico afhankelijk is van de ontwikkeling van IBM Watson. Of het implementeren van een volledig functionerend prototype te veel werk is zal blijken uit het onderzoeksverslag, opgesteld in de onderzoeksfase.

Reactie Wanneer blijkt dat het realiseren van een prototype zodanig veel werk met zich meebrengt dat dit niet binnen de afstudeerperiode ontwikkeld kan worden, dan zal in ieder geval worden gedocumenteerd wat gerealiseerd moet worden (dit gebeurt in het onderzoeksverslag). De realisatie van het prototype hangt, zoals eerder aangegeven, af van het onderzoek. Na afloop van het onderzoek zal worden besloten welke onderdelen binnen het tijdsbestek van de afstudeerperiode ontwikkeld kunnen worden. Daarna zal per week een sprint worden opgesteld waarin wordt bestolen wat in die week wordt ontwikkeld.

- **De onderzoeksfase blijkt meer tijd nodig te hebben dan in eerste instantie verwacht.**

Preventie Om dit te voorkomen is per deelvraag een ruime tijds marge genomen, wanneer deze marge niet wordt gehaald zal dit ingehaald worden gedurende de tijds marge van een volgende deelvraag.

Reactie Wanneer de onderzoeksfase niet volledig is afgerond op het geplande moment zal de ontwikkelfase later beginnen. Dit is op zich geen probleem aangezien de maand juni vrij wordt gehouden voor eventuele uitloop.

- **Ondanks de beschikbare trainingsdata is Watson niet in staat om vragen vanuit de verzekeringsbranche op een juiste manier te kunnen beantwoorden.**

Preventie Omdat Watson gebruik maakt van machine learning wat onverwachte resultaten kan opleveren is dit niet volledig uit te sluiten. Er zal in ieder geval trainingsdata worden geleverd zodanig dat aan de kwaliteitseisen voor trainingsdata wordt voldaan. Deze eisen zijn opgesteld door IBM en worden tijdens de onderzoeksfase in het onderzoeksverslag gedefinieerd.

Reactie Wanneer Watson niet voldoende getraind blijkt te zijn zal in eerste instantie worden gekeken of dit op te lossen is door meer trainingsmateriaal te leveren. Als dit niet het geval is zal er waarschijnlijk iets mis zijn gegaan tijdens het trainingsproces. In dat geval zal contact worden opgenomen met IBM om te bespreken hoe dit probleem verholpen kan worden. Deze service is beschikbaar aangezien Colimbra een partner is geworden van het IBM Watson ecosysteem (zoals beschreven in paragraaf 3.3).

7. Tijdlijn

Dit hoofdstuk zal de tijdlijn omschrijven. Hierin worden onder andere de belangrijke opleveringsdata genoemd. In paragraaf 7.1 wordt aangegeven welke data belangrijk zijn voor de oplevering van schooldocumentatie en welke overige gerelateerde zaken op welk tijdstip worden uitgevoerd. De paragrafen 7.2 en 7.3 geven hetzelfde aan voor respectievelijk de onderzoeksfase en de ontwikkelfase. Uiteindelijk zal in paragraaf 7.4 een totaalbeeld worden gegeven.

De tijdlijn hieronder zal zo goed mogelijk gevolgd worden. Het kan echter voorkomen dat een aantal onderdelen door elkaar lopen door informatie wat verschaft wordt via conference calls met IBM. Zo is bijvoorbeeld tegen de planning in begonnen met het verzamelen van informatie over deelvraag drie. Dit omdat al vroeg in de onderzoeksfase een kick-off conference call gepland stond met IBM waarin deze achtergrondinformatie gewenst was.

7.1 Schoolzaken

De onderstaande tabel geeft aan welke inleverdata voor schooldocumenten worden gehanteerd tijdens de afstudeerperiode.

Datum	Document/taak	Status	Tijd
Januari			
15-01-2016	Afstudeervoorstel	Goedgekeurd	-
25-10-2016	Leerteam sessie 1: kick-off		3u
Februari			
12-02-2016	Plan van aanpak	1 ^e concept	3d
16-02-2016	1 ^e afspraak docentbegeleider		1u
19-02-2016	Leerteam sessie 2: plan van aanpak bespreking		1.5u
29-02-2016	Plan van aanpak	2 ^e concept	5d
Maart			
11-03-2016	Plan van Aanpak	Definitief	3d
18-03-2016	Afstudeercontract	Ondertekend	1u
28-03-2016	Scriptie	1 ^e concept	25u
April			
-	Leerteam sessie 3: scriptbespreking		1.5u
25-04-2016	Scriptie	2 ^e concept	20u
Mei			
31-05-2016	Scriptie	Definitief	36u
Juni			
-	Uitloop/extra functionaliteit		22d
-	Afstudeerzitting		-

De volgende zaken worden in de loop van de afstudeerperiode, via e-mail, ingepland:

- Leerteam sessie - Sessie 3: Script bespreking (begin april)
- Afstudeerzitting (7 t/m 17 juni 2016)

7.2 Onderzoeksfase

De onderzoeksfase loopt in februari en maart. De onderstaande tabel geeft aan welke data in de onderzoeksfase wordt gehanteerd.

Datum	Document/taak	Status	Tijd
Februari			
01-02-2016	Start onderzoeksfase		-
09-02-2016	Kick-off Colimbra-IBM Watson		1u
19-02-2016	Deelvraag 1 beantwoord & verwerkt		3d
26-02-2016	Deelvraag 2 beantwoord & verwerkt		10d
29-02-2016	Onderzoeksverslag	1 ^e concept	13d
Maart			
04-03-2016	Deelvraag 4 beantwoord & verwerkt		4d
11-03-2016	Deelvraag 5 beantwoord & verwerkt		4d
25-03-2016	Deelvragen 6 en 7 beantwoord & verwerkt		10d
31-03-2016	Deelvraag 8 beantwoord & verwerkt		2d
31-03-2016	Onderzoeksverslag	Definitief	20d
31-03-2016	Afronding onderzoeksfase		-

7.3 Ontwikkelfase

De ontwikkelfase zal lopen vanaf april t/m juni. Dit is een maand langer dan de onderzoeksfase. In april en mei zal echter ook tijd worden vrijgemaakt voor de afstudeerscriptie (zie paragraaf 7.1). De verschillende sprints zijn gepland in april en mei, hierdoor kan juni worden gebruikt als uitlooptijd of voor eventuele extra functionaliteit.

Datum	Document/taak	Status	Tijd
April			
01-04-2016	Start ontwikkelfase		-
03-04-2016	Globale planning ontwikkelfase		1d
04-04-2016	Start sprint 1 in overleg met begeleider		5d
11-04-2016	Start sprint 2 in overleg met begeleider		5d
18-04-2016	Start sprint 3 in overleg met begeleider		5d
25-04-2016	Start sprint 4 in overleg met begeleider		5d
Mei			
02-05-2016	Start sprint 5 in overleg met begeleider		5d
09-05-2016	Start sprint 6 in overleg met begeleider		5d
16-05-2016	Start sprint 7 in overleg met begeleider		5d
23-05-2016	Start sprint 8 in overleg met begeleider		5d
31-05-2016	Prototype	1 ^e versie	40d
Juni			
-	Uitloop/extra functionaliteit		22d
30-06-2016	Prototype	Definitief	22d
30-06-2016	Afronding ontwikkelfase		-

Zoals in de bovenstaande tabel te zien is zal begin april een abstracte invulling gegeven worden aan de sprints, per week zal deze worden uitgewerkt en indien nodig aangepast.

7.4 Totaalbeeld

De onderstaande tabel is een samenvoeging van alle in de vorige paragrafen vermelde data.

Datum	Document/taak	Status	Tijd
Januari			
15-01-2016	Afstudeervoorstel	Goedgekeurd	-
25-10-2016	Leerteam sessie 1: kick-off		3u
Februari			
01-02-2016	Start onderzoeksfase		-
09-02-2016	Kick-off Colimbra-IBM Watson		1u
12-02-2016	Plan van aanpak	1 ^e concept	3d
19-02-2016	Deelvraag 1 beantwoord & verwerkt		3d
16-02-2016	1 ^e afspraak docentbegeleider		1u
19-02-2016	Leerteam sessie 2: plan van aanpak bespreking		1.5u
26-02-2016	Deelvraag 2 beantwoord & verwerkt		10d
29-02-2016	Onderzoeksverslag	1 ^e concept	13d
29-02-2016	Plan van aanpak	2 ^e concept	5d
Maart			
04-03-2016	Deelvraag 3 beantwoord & verwerkt		4d
11-03-2016	Deelvraag 4 beantwoord & verwerkt		4d
18-03-2016	Plan van Aanpak	Definitief	3d
18-03-2016	Afstudeercontract	Ondertekend	1u
25-03-2016	Deelvragen 5 en 6 beantwoord & verwerkt		10d
28-03-2016	Scriptie	1 ^e concept	25u
31-03-2016	Deelvraag 7 beantwoord & verwerkt		2d
31-03-2016	Onderzoeksverslag	Definitief	20d
31-03-2016	Afronding onderzoeksfase		-
April			
01-04-2016	Start ontwikkelfase		-
-	Leerteam sessie 3: scriptbespreking		1.5u
03-04-2016	Globale planning ontwikkelfase		1d
04-04-2016	Start sprint 1 in overleg met begeleider		5d
11-04-2016	Start sprint 2 in overleg met begeleider		5d
18-04-2016	Start sprint 3 in overleg met begeleider		5d
25-04-2016	Start sprint 4 in overleg met begeleider		5d
25-04-2016	Scriptie	2 ^e concept	20u
Mei			
02-05-2016	Start sprint 5 in overleg met begeleider		5d
09-05-2016	Start sprint 6 in overleg met begeleider		5d
16-05-2016	Start sprint 7 in overleg met begeleider		5d
23-05-2016	Start sprint 8 in overleg met begeleider		5d
31-05-2016	Prototype	1 ^e versie	40d
31-05-2016	Scriptie	Definitief	36u
Juni			
-	Uitloop/extra functionaliteit		22d
-	Afstudeerzitting		-
30-06-2016	Prototype	Definitief	22d
30-06-2016	Afronding ontwikkelfase		-

8. Contactgegevens

Op de titelpagina zijn de algemene contactgegevens al vermeld. Voor de volledigheid zijn die hieronder herhaald met toevoeging van de contactgegevens van de, voor deze afstudeerperiode, relevante hoofdpersonen. Net zoals op de titelpagina wordt ook in dit hoofdstuk gebruik gemaakt van afkortingen, de betekenissen hiervan staan hieronder vermeld.

- S = Studentnummer;
- T = Telefoonnummer;
- E = E-mailadres;
- F = Functie.

Paragraaf 8.1 bevat de contactgegevens van de afstudeerder, paragraaf 8.2 geeft de contactgegevens van de relevante personen vanuit Hogeschool Utrecht, paragraaf 8.3 de gegevens van relevante personen vanuit Colimbra en uiteindelijk geeft paragraaf 8.4 de relevante contactgegevens vanuit IBM.

8.1 Afstudeerder

Onderstaand de contactgegevens van de afstudeerder.

Rick Koster

Prinses Beatrixhof 10
1411CK Naarden

S: 1622528



E: rick.koster@student.hu.nl

E: rkoster@colimbra.com

F: Afstudeerder

8.2 Hogeschool Utrecht

De onderstaande contactgegevens zijn van de relevante personen vanuit Hogeschool Utrecht. Henk van Nimwegen zal fungeren als docentbegeleider en is daarom een vast aanspreekpunt.

Hogeschool Utrecht

Nijenoord 1
3552 AS Utrecht

T: 088 481 82 83

E: info@hu.nl

Henk van Nimwegen

F: Docentbegeleider

F: 2^e examiner

Joop Kaldeway

F: 1^e examiner

Marten Wensink

F: Afstudeercoördinator

8.3 Colimbra

Onderstaand de contactgegevens van de relevante hoofdpersonen vanuit Colimbra. Ronald van den Berg zal fungeren als bedrijfsbegeleider tijdens de afstudeerperiode. Bram van Doveren en Jacco van Leeuwen zijn beide net zoals Ronald directeurs van Colimbra. Bram en Jacco zijn echter meer van de business kant en zullen daarom geen technische begeleiding bieden maar zullen wel meegenomen worden in de gemaakte beslissingen.

Colimbra B.V.

Kruisweg 821
2132 NG Hoofddorp

T: 023 565 37 38

E: info@colimbra.com

Ronald van den Berg

F: Bedrijfsbegeleider
F: Directeur

Bram van Doveren

F: Directeur

Jacco van Leeuwen

F: Directeur

8.4 International Business Machines (IBM)

Tijdens de afstudeerperiode zal regelmatig contact worden opgenomen met IBM (vanuit Amerika). Joshua Hall is de "relationship manager" en zal daarom als vast aanspreekpunt fungeren voor vragen gericht aan IBM.

Joshua H. Hall

F: Ecosystem Relationship
Manager

9. Afkortingen en begrippen

Dit hoofdstuk geeft een, op alfabet gesorteerde, opsomming van afkortingen en begrippen die gebruikt zijn binnen dit document. Paragraaf 9.1 omschrijft de afkortingen en paragraaf 9.2 omschrijft de begrippen.

9.1 Afkortingen

AI

Artificial Intelligence (kunstmatige intelligentie)

API

Application Programming Interface

IBM

International Business Machines

PvA

Plan van aanpak

REST

Representational State Transfer

STD

State Transition Diagram

9.2 Begrippen

Application Programming Interface: Een interface waarmee met behulp van programmeercode een bepaalde applicatie/service benaderd kan worden.

Artificial Intelligence: De Engelse benaming voor kunstmatige intelligentie. Wanneer gesproken wordt over een AI wordt vaak een supercomputer bedoeld met menselijke intelligentie.

Bluemix: Een cloud platform van IBM waar applicaties in gehost kunnen worden. Deze applicaties hebben via Bluemix toegang tot verschillende IBM-services.

Google hangouts: Een oplossing om conference calls te houden met behulp van voice over IP. Kan vergeleken worden met een groepsgesprek in Skype.

International Business Machine (IBM): Een Amerikaans bedrijf dat zich bezighoudt met de ontwikkeling van hardware, middleware en software, en het bieden van infrastructuur, hosting en consultancy services. Besproken in hoofdstuk 3

Klassendiagram: Wordt opgesteld om een overzichtelijke weergave te geven van de verschillende, in code opgenomen, klassen achter een applicatie. Hierin is ook te zien hoe welke klassen met elkaar in verband staan (Ambler, 2004).

MoSCoW methode: Omschrijft de "must", "should", "could" en "won't" have onderdelen van een applicatie. Dit houdt in dat wordt omschreven welke onderdelen de applicatie moet hebben om te kunnen functioneren, welke onderdelen de applicatie zou moeten hebben, welke onderdelen

eventueel toegevoegd kunnen worden en uiteindelijk welke onderdelen niet in de applicatie terecht komen (Stapleton, 2003).

Natuurlijke taal: Een natuurlijke taal of etnische taal is in de taal filosofie elke levende of dode taal die door een aantal mensen als hun moedertaal wordt of werd beschouwd. Taal waarvan de regels zijn afgeleid uit het gangbare gebruik i.p.v. op formele wijze te zijn opgesteld, zoals bij een programmeertaal.

Proof of concept: Een proof of concept is een prototype wat ontwikkeld is om een mogelijkheid (concept) van een applicatie te bewijzen.

REST-API: Representational state transfer application programming interface. Dit houdt in dat de services die gebruik maken van REST-API te benaderen zijn via webrequests. Deze services worden ook wel RESTful services genoemd (Massé, 2012).

SCRUM: Een iteratieve Agile werkmethode waarbij per week zogenaamde "sprints" worden opgesteld. Deze sprints zijn de mijlpalen van het project (Overheem, 2004).

Sequentiediagram: Een diagram waarin wordt getoond hoe verschillende services met een applicatie communiceren (Ambler, 2004).

State transition diagram (STD): uitgewerkt om aan te geven in welke state een applicatie zich bevindt en welke andere states beschikbaar zijn vanuit de huidige state (Ambler, 2004).

Use-case diagram: Bedoeld om met behulp van een diagram duidelijk te geven over het gebruik van een applicatie. Hierin is te zien welke gebruikers er zijn en hoe deze gebruikers gebruik maken van de applicatie (Maßen & Lichter, 2002).

Verzekeringspolis: Een persoonlijk document vanuit de verzekeringsbranche waarin wordt aangegeven welke afspraken zijn gemaakt met de verzekeringsmaatschappij.

Verzekeringspolisvoorwaarden: Een algemeen document vanuit de verzekeringsbranche, hierin worden de voorwaarden aan een polis document vastgelegd.

Watson: Een supercomputer ontwikkeld door International Business Machines (IBM). Deze supercomputer is in staat om antwoorden te geven op vragen die gesteld zijn in natuurlijke taal. Een uitgebreidere omschrijving van IBM Watson is gegeven in hoofdstuk 3.

10. Bibliografie

- Ambler, S. (2004). *The Object primer: Agile Model-Driven Development with UML 2.0*. Cambridge University Press. Opgehaald van <http://www.ambysoft.com/books/theObjectPrimer.html>
- Apello, J. (2008). *Management 3.0 – Leading Agile Developers*. Boston: Addison-Wesley.
- Berg, R. v. (2016). (R. Koster, Interviewer) Hoofddorp.
- Best, J. (2015). *Tech Republic*. Opgeroepen op 02 15, 2016, van IBM Watson: The inside story of how the Jeopardy-winning supercomputer was born, and what it wants to do next: <http://www.techrepublic.com/article/ibm-watson-the-inside-story-of-how-the-jeopardy-winning-supercomputer-was-born-and-what-it-wants-to-do-next/>
- Campbell, M. (2002). *Artificial Intelligence*. USA: Elsevier. Opgehaald van http://ac.els-cdn.com/S0004370201001291/1-s2.0-S0004370201001291-main.pdf?_tid=7b4df6d0-d61b-11e5-932a-00000aab0f6c&acdnat=1455785102_5325cb0fccff072f174de5179e2acb76
- Hogeschool Utrecht. (2015, 10 06). *Afstudeerleidraad Instituut voor ICT*. Faculteit Natuur en Techniek, Utrecht. Opgehaald van <http://www.voorbedrijven.hu.nl/~media/hu-bedrijven/docs/praktijkbureau%20cluster%20ict/afstudeerleidraad%20instituut%20voor%20ict%20cursus%202015-2016.pdf?la=nl>
- IBM. (2016). *IBM Bluemix*. Opgeroepen op 02 10, 2016, van IBM Bluemix: <http://www.ibm.com/cloud-computing/bluemix/>
- IBM. (2016). *IBM Watson*. Opgeroepen op 02 02, 2016, van IBM: <http://www.ibm.com/smarterplanet/us/en/ibmwatson/index.html>
- IBM. (2016). *IBM Watson Gallery*. Opgeroepen op 02 15, 2016, from IBM Watson Developer Cloud: <http://www.ibm.com/smarterplanet/us/en/ibmwatson/developercloud/gallery.html>
- IBM. (2016). *Watson Services*. Opgeroepen op 02 09, 2016, van IBM Watson Developer Cloud: <https://www.ibm.com/smarterplanet/us/en/ibmwatson/developercloud/services-catalog.html>
- Kempen, G. (1987). *Natuurlijke taal en kunstmatige intelligentie: taal tussen mens en machine*. Opgehaald van http://pubman.mpg.de/pubman/item/escidoc:1244615/component/escidoc:1368653/Kempen_Levelt--Mens_en_Computer_1987.pdf
- Kleeff, I. v. (2015). *IBM Watson businesscase - Colimbra*. Hoofddorp.
- Massé, M. (2012). *REST API Design Rulebook*. United States: O'Reilly Media. Opgeroepen op 02 02, 2016, van https://books.google.nl/books?hl=en&lr=&id=eABpzyTcJNIC&oi=fnd&pg=PR3&dq=REST+api&ots=vyTB--maKz&sig=PD9L8rZsb9qavNszeUPrVU_8_Os#v=onepage&q=REST%20api&f=false

- Maßen, T. v., & Lichter, H. (2002). *Modeling Variability by UML Use Case Diagrams*. Essen.
Opgehaald van
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.20.2206&rep=rep1&type=pdf#page=25>
- Overheem, B. (2004). *Scrum*. Opgehaald van Wat is SCRUM?: <http://www.scrum.nl/site/Watis-Scrum-agile-scrum>
- Poecke, L. V. (1991). *Verbale communicatie*. Opgehaald van
<https://books.google.nl/books?id=w0nEDMvnBncC&pg=PA13&lpg=PA13&dq=natuurlijke+taal+definitie&source=bl&ots=4fbeKYeNkJ&sig=bLVs9pNTosdmFgeweJXZ7aMlqnE&hl=en&sa=X&ved=0ahUKEwinqonBzZzLAhWBE5oKHbaJCJAQ6AEINzAE#v=onepage&q=natuurlijke%20taal%20definitie&f=false>
- Ravencroft, J. (2016). Conference call: Colimbra & Watson Technical Session.
- Stapleton, J. (2003). *DSDM: Business Focused Development*. Boston.

11. Bijlage

11.1 Bijlage A: Interview – Ronald van den Berg

Projectgrenzen

Waar wordt geen onderzoek naar gedaan?

- Geen onderzoek naar de interne werking van Watson.
 - De manier waarop natuurlijke taal wordt ontleed (geparsed);
 - De manier waarop Watson leert. Het gaat hierbij om de gebruikte machine learning technieken, wel wordt onderzocht hoe Watson ingesteld/geprogrammeerd kan worden om te leren van interacties met gebruikers.

Wat zal niet zelf worden geïmplementeerd?

- De applicatie een antwoord laten formuleren in natuurlijke taal (op dit moment lijkt het erop dat dit niet ondersteund wordt vanuit IBM Bluemix, het antwoord is vooraf gedefinieerd of is een quote uit een document).
- Er wordt geen aandacht besteed aan het stijlen van een userinterface. Dit project betreft een proof of concept om de werking van Watson aan te tonen, een klantvriendelijke interface zal in een ander stadium worden ontwikkeld. Wel zal een eenvoudige interface, mogelijk zonder uitgebreide styling, worden gemaakt om gedurende de ontwikkeling met Watson te kunnen communiceren.

Wat wordt eventueel zelf geïmplementeerd, als IBM geen oplossing biedt?

- Een wrapper/SDK voor Watson services in c# .NET. IBM heeft een opensource Java SDK beschikbaar.

Ontwerp

Zijn er specifieke modellen/methoden gewenst in het functioneel ontwerp?

- MoSCoW methode (prioriteiten)
- Use-case diagram (gebruikssituatie)

Zijn er specifieke modellen/methoden gewenst in het technisch ontwerp?

- Klassendiagram (applicatiestructuur)
- Sequentiediagram (applicatieverloop)
- State Transition Diagram (applicatieverloop)

Onderzoeksverslag**Worden de applicaties uiteindelijk gehost via Bluemix?**

Nee, Bluemix kan gebruikt worden voor prototyping. De uiteindelijke applicatie zal echter niet in Bluemix gehost worden.

Zijn er aanvullingen-aanpassingen op de onderstaande MoSCoW methode?

MoSCoW		Watson service
Must	Dialoog om de gebruiker te begeleiden	Dialog
	Mogelijkheid om polis documenten en polisvoorwaarden te uploaden	Document Conversion
	Informatie uit beschikbaar gestelde documenten moet te benaderen zijn via de dialoog.	Retrieve & Rank
Should	Beter begrip van wat de gebruiker wil bereiken (de dialoog service let alleen op zinsopbouw met behulp van keywords. Met deze toevoeging zal de intentie van de input worden begrepen waardoor beter gereageerd kan worden).	Natural Language Classifier
Could	Gesproken aansturing	Speech to Text en Text to Speech
	Admin interface om Watson te trainen	-
Won't	Ondersteuning voor de Nederlandse taal (op dit moment niet ondersteund door Watson).	
	Socialmedia analyse om informatie over de gebruiker te verzamelen	AlchemyAPI en/of Personality Insights
	Object herkenning om geüploade plaatjes te begrijpen	Visual recognition

Kwaliteitscriteria**Onderzoeksfase**

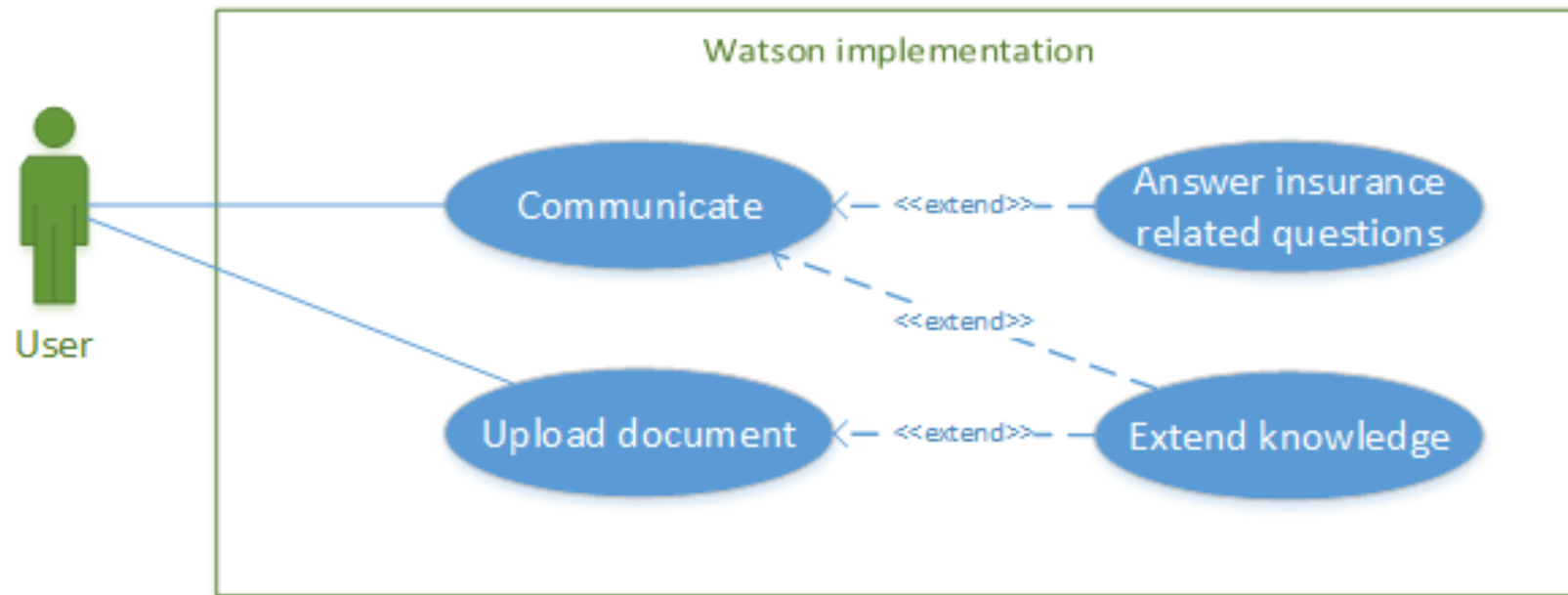
Moet duidelijkheid bieden over de werking van Watson, specifiek gericht op de use-case.

Ontwikkelfase

In ieder geval de must en should have onderdelen uit de MoSCoW methode moeten ontwikkeld zijn.

Bijlage B: Functioneel ontwerp

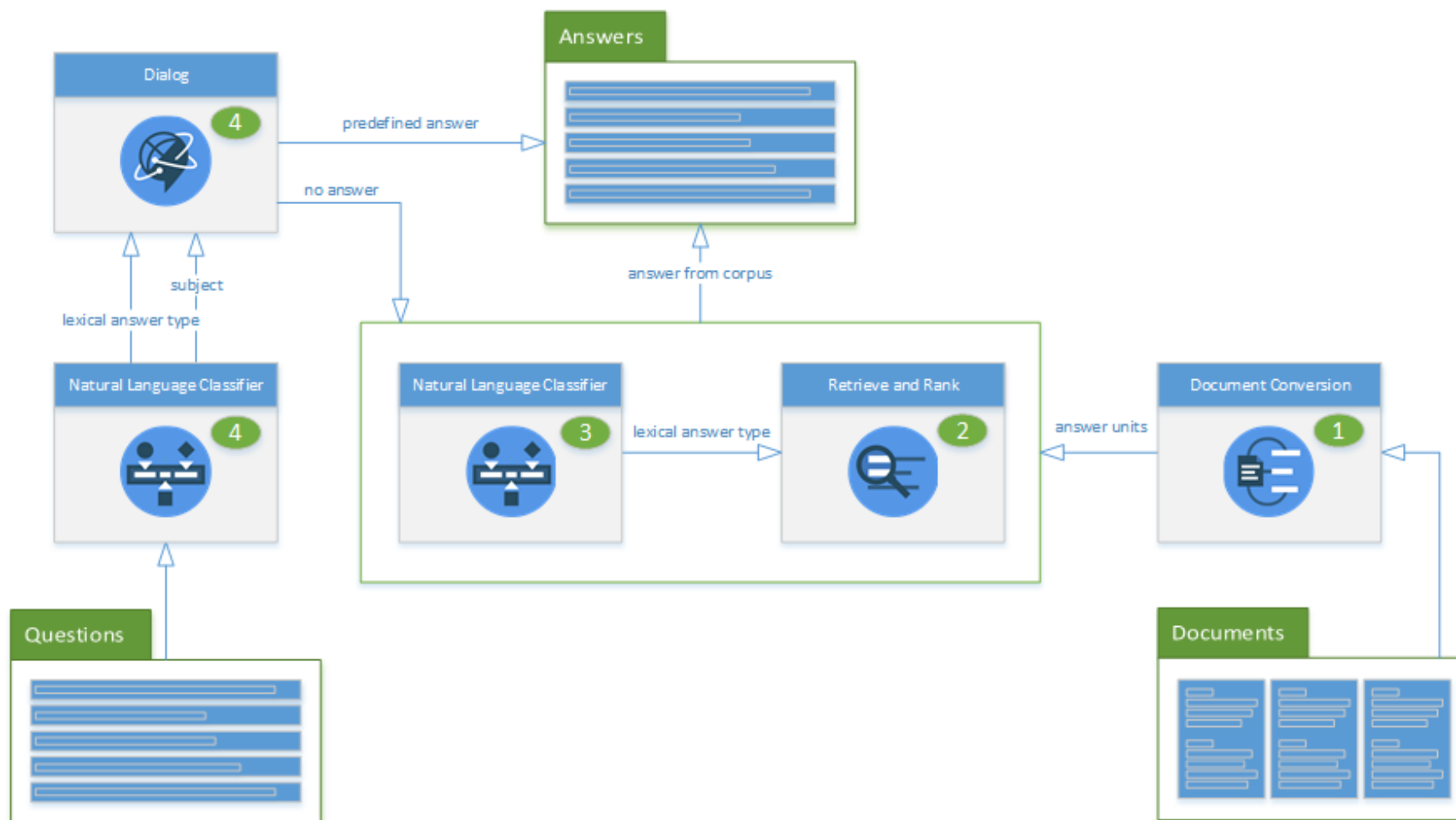
Use case diagram



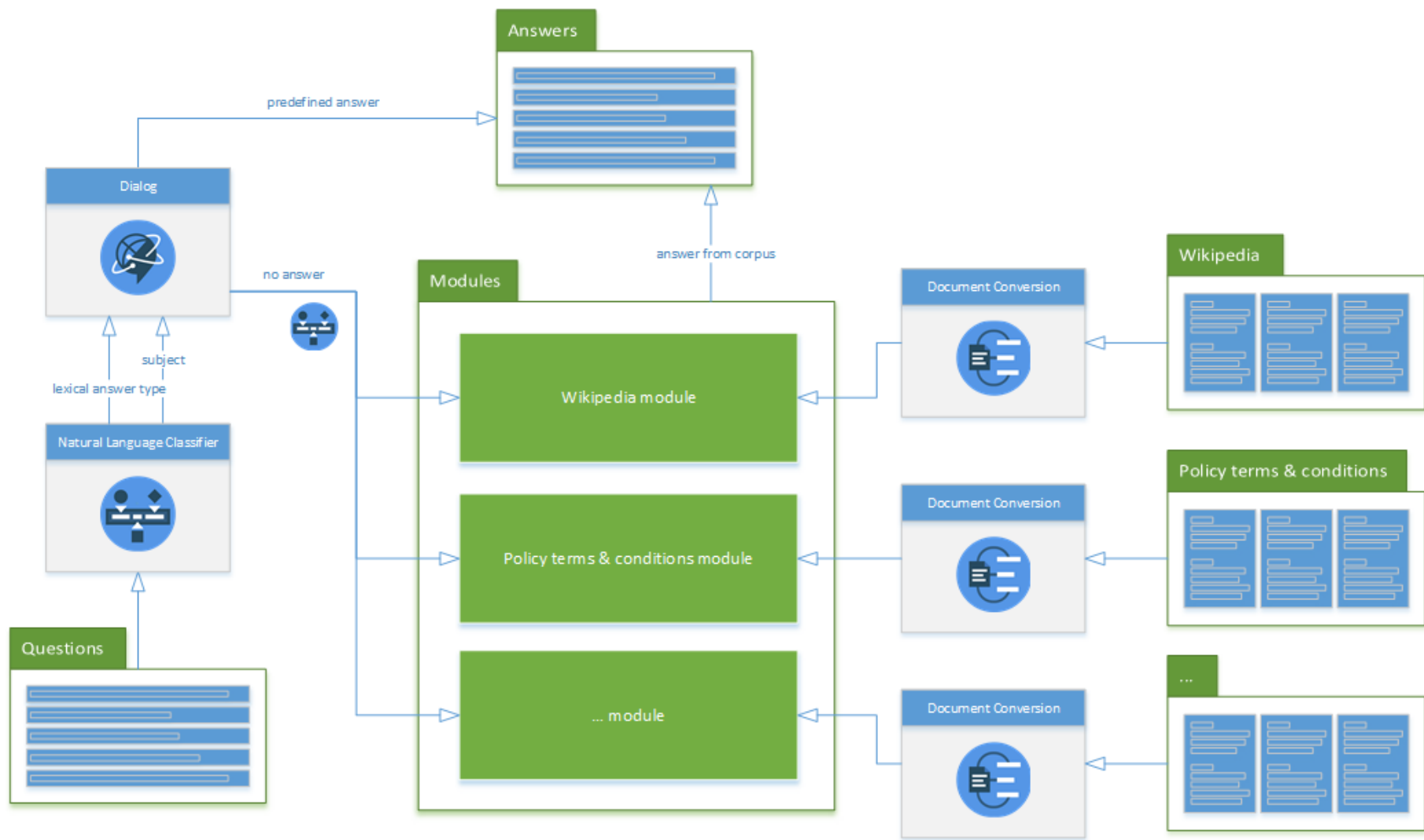
MoSCoW method

MoSCoW		Watson service(s)
Must	Dialoog om de gebruiker te begeleiden	Dialog
	Mogelijkheid om polisdocumenten en polisvoorwaarden te uploaden	Document Conversion
	Informatie uit beschikbaar gestelde documenten moet te benaderen zijn via de dialoog.	Retrieve & Rank
Should	Beter begrip van wat de gebruiker wil bereiken (de dialoog service let alleen op zinsopbouw met behulp van keywords. Met deze toevoeging zal de intentie van de input worden begrepen waardoor beter gereageerd kan worden).	Natural Language Classifier
	Chatinterface om Watson te benaderen.	-
Could	Gesproken aansturing	Speech to Text en Text to Speech
	Admin interface om Watson te trainen	-
	Het bijhouden van gebruik in een database om Watson op een later stadium verder te kunnen trainen.	-
Won't	Ondersteuning voor de Nederlandse taal (op dit moment niet ondersteund door Watson).	-
	Socialmedia analyse om informatie over de gebruiker te verzamelen	AlchemyAPI en/of Personality Insights
	Object herkenning om geüploade plaatjes te begrijpen	Visual recognition

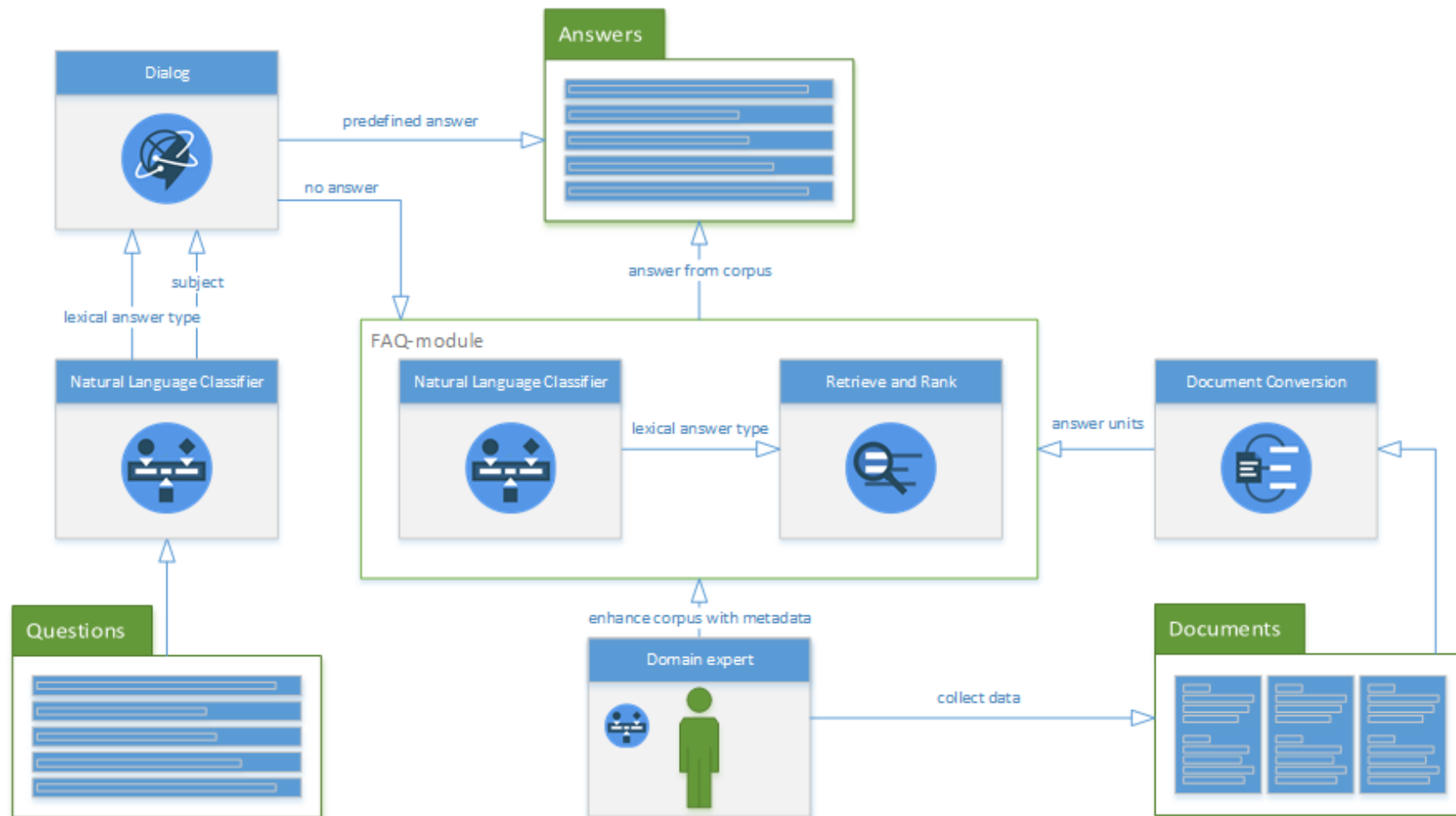
Bijlage C: Technisch ontwerp



Architecture - core



Architecture - extended



Architecture - prototype

<<Interface>>

ITelegramBot

<<event>>

+ OnStart : OnStartEventHandler

<<event>>

+ OnStop : OnStopEventHandler

+ Id : long

+ FirstName : string

+ LastName : string

+ Username : string

+ ViewName : string

+ SetToken(string) : ITelegramBot

+ SetPollingDelay(TimeSpan) : ITelegramBot

+ Start() : ITelegramBot

+ Stop() : ITelegramBot

+ AddUpdate(Update) : ITelegramBot

+ SetWebhookAsync(string, [InputFile]) : Task<ITelegramBot>

+ RemoveWebhookAsync() : Task<ITelegramBot>

+ GetMeAsync() : Task<User>

+ SendMessageAsync(Chat, string, [...]) : Task<Message>

+ SendCodeAsync(Chat, string, [...]) : Task<Message>

+ ForwardMessageAsync(Chat, Message, [...]) : Task<Message>

+ SendPhotoAsync(Chat, Photo, [...]) : Task<Message>

+ SendAudioAsync(Chat, object, [...]) : Task<Message>

+ SendDocumentAsync(Chat, string, [...]) : Task<Message>

+ SendStickerAsync(Chat, InputFile, [...]) : Task<Message>

+ EditMessageTextAsync(Chat, Message, string, [...]) : Task<Message>

+ EditCodeTextAsync(Chat, Message, string, [...]) : Task<Message>

+ DownloadFileAsync(Document) : Task<InputFile>

Telegram Bot

- updateQueue : ConcurrentQueue<Update>

- chatTasks : ConcurrentDictionary<Chat, Task>

Token : string

TelegramBot()

PeekUpdate(Chat, User, Update) : Task<bool>

UpdateReceived(Chat, User, Update) : Task

CommandReceived(Chat, User, Command) : Task

MessageReceived(Chat, User, Message) : Task

CallbackQueryReceived(Chat, User, CallbackQuery) : Task

OnErrorReceived(TelegramException) : void

- GetUpdatesAsync() : Task<List<Update>>

- RequestAsync<T>(<string, [Dictionary<string, object>]>) : Task<T>

TelegramBot<T>

- datasets : ConcurrentDictionary<Chat, T>

PeekUpdate(Chat, User, Update, T) : Task<bool>

UpdateReceived(Chat, User, Update, T) : Task

CommandReceived(Chat, User, Command, T) : Task

MessageReceived(Chat, User, Message, T) : Task

CallbackQueryReceived(Chat, User, CallbackQuery, T) : Task

Colimbra.Telegram.Api

Watson.Telegram.Api

TelegramBot<T>

ColimbraBot : TelegramBot<ColimbraBotData>

```

- chatAuthorization : List<long>
- commandAuthorization : List<long>
- developmentMode : bool
# PeekUpdate(Chat, User, Update, ColimbraBotData) : Task<bool>
# UpdateReceived(Chat, User, Update, ColimbraBotData) : Task
# CommandReceived(Chat, User, Command, ColimbraBotData) : Task
# MessageReceived(Chat, User, Message, ColimbraBotData) : Task
# CallbackQueryReceived(Chat, User, CallbackQuery, ColimbraBotData) : Task

- InitializeAsync(Chat, User, ColimbraBotData) : Task

```

ColimbraBotData

```

+ WatsonUser : Watson.Core.WatsonUser
+ LoadingMessage : Telegram.Api.Message

+ Initialized : bool
+ ShowJson : bool
+ ShowSummary : bool
+ TrainMode : bool

+ RequireParameter : bool
+ QueuedCommand : Telegram.Api.Command
+ QueuedService : Watson.Api.IWatsonService
+ QueuedMethod : System.Reflection.MethodInfo
+ RequiredParameters : List<System.Reflection.ParameterInfo>
+ Parameters : List<object>

```

Colimbra.Watson.Core

Dialog

DocumentConversion

NaturalLanguageClassifier

RetrieveAndRank

Dialog

```

+ ListDialogsAsync() : Task<DialogCollection>
+ CreateDialogAsync(string, InputFile) : Task<Dialog>
+ UpdateDialogAsync(string, InputFile) : Task<EmptyModel>
+ GetDialogAsync(string) : Task<Dialog>
+ DeleteDialogAsync(string) : Task<EmptyModel>

+ GetContentAsync(string) : Task<ContentCollection>
+ UpdateContentAsync(string, string, string) : Task<EmptyModel>

+ Converse(string, string, [string], [string]) : Task<Conversation>

+ GetProfileAsync(string, string, [string]) : Task<ProfileCollection>
+ UpdateProfileAsync(string, string, string) : Task<ProfileCollection>

```

DocumentConversion

```

+ ConvertCustomAsync(InputFile, InputFile) : Task<Conversation>
+ ConvertAsync(InputFile) : Task<Conversation>

```

NaturalLanguageClassifier

```

+ ListClassifiersAsync() : Task<ClassifierCollection>
+ CreateClassifierAsync(string, InputFile) : Task<Classifier>
+ GetClassifierAsync(string) : Task<Classifier>
+ DeleteClassifierAsync(string) : Task<EmptyModel>

+ ClassifyAsync(string, string) : Task<Classification>

```

RetrieveAndRank

```

- ranker : Api.Ranker
- cluster : Api.Cluster
+ ListClustersAsync() : Task<ClusterCollection>
+ CreateClusterAsync(string, string) : Task<Cluster>
+ GetClusterAsync(Cluster) : Task<Cluster>
+ DeleteClusterAsync(Cluster) : Task<EmptyModel>

+ ListConfigurationsAsync(Cluster) : Task<ConfigurationCollection>
+ CreateConfigurationAsync(Cluster, string, InputFile) : Task<Configuration>
+ GetConfigurationAsync(Cluster, Configuration) : Task<Configuration>
+ DeleteConfigurationAsync(Cluster, string) : Task<EmptyModel>

+ ListCollectionsAsync(Cluster) : Task<CollectionCollection>
+ CreateCollectionAsync(Cluster, Configuration, string) : Task<EmptyModel>
+ DeleteCollectionAsync(Cluster, string) : Task<EmptyModel>

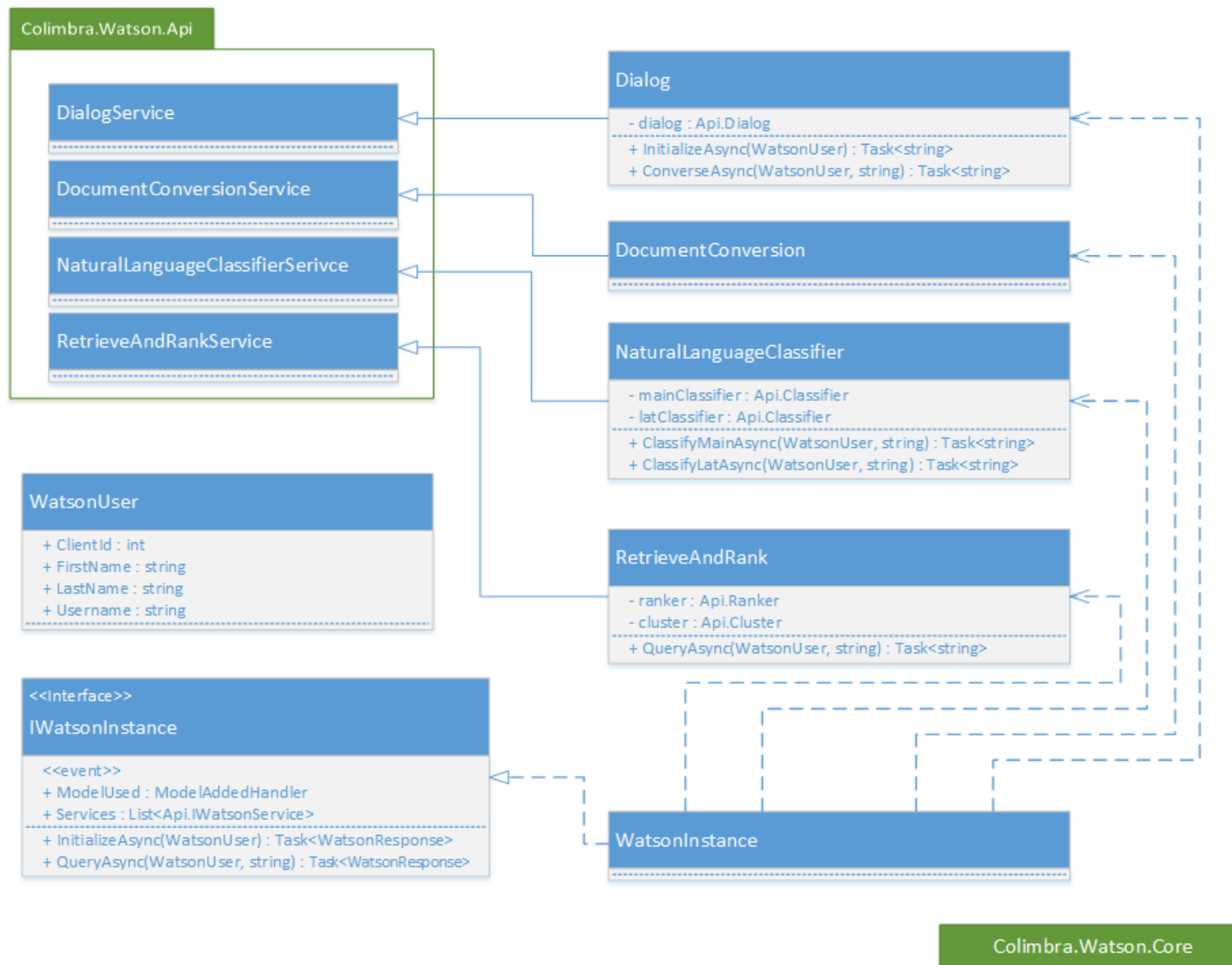
+ SelectAsync(Cluster, string, string, [List<string>]) : Task<SolrResponse>
+ FcSelectAsync(Cluster, string, Ranker, string, [List<string>]) : Task<SolrResponse>

+ ListRankersAsync() : Task<RankerCollection>
+ CreateRankerAsync(Cluster, string, string, InputFile) : Task<Ranker>
+ GetRankerAsync(Ranker) : Task<Ranker>
+ DeleteRankerAsync(Ranker) : Task<EmptyModel>
+ RankAsync(Ranker, InputFile) : Task<Rank>

```

Colimbra.Telegram.ColimbraBot





Bijlage D: Watson serviceoverzicht

1. AlchemyAPI

De AlchemyAPI service is een set van drie services waarmee ontwikkelaars cognitieve applicaties kunnen maken. Met deze services kunnen de inhoud en context van tekst en plaatjes door het systeem begrepen worden.

- **Alchemy language**

De alchemy language service is een collectie van 12 API's waarmee tekstanalyse van natuurlijke taal kan wordt gerealiseerd. Het gaat hierbij om de volgende API's:

- Sentiment analysis;
- Concept tagging;
- Entity extraction;
- Keyword extraction;
- Relation extraction;
- Author extraction;
- Authors extraction;
- Language detection;
- Micro formats parsing;
- Date extraction;
- Emotion analysis;
- Publication date.

- **Alchemy vision**

De alchemy vision service begrijpt en analyseert complexe visuele scènes zonder tekstuele hints nodig te hebben. De volgende API's worden aangeboden:

- Face detection;
- Image tagging.

- **Alchemy data**

De alchemy data service analyseert nieuws en blog artikelen door gebruik te maken van gerichte zoek algoritmen en een trendanalyse. Alchemy data bevat de volgende API's:

- News API (nieuws artikelen en blogs);
- Taxonomy API (het categoriseren van webpagina's);
- Text extraction;
- Image link extraction;
- Feed detection.

Demo <http://www.alchemyapi.com/products/demo>

Colimbra

De AlchemyAPI service kan een grote hoeveelheid informatie verzamelen uit tekst en plaatjes. Het verzamelen van deze informatie is belangrijk voor het prototype maar hiervoor wordt geen gebruik gemaakt van deze services. Dit omdat de AlchemyAPI services vooral zijn gericht op het analyseren van tekst afkomstig uit nieuws gerelateerde webpagina's. Voor het extraheren van de benodigde informatie uit tekst worden voor het prototype andere services gebruikt (Natural Language Classifier & Retrieve and Rank).

2. Concept Expansion

De Concept Expansion service is bedoeld om grote hoeveelheden tekst te analyseren en hierbij een woordenboek van synoniemen op te stellen. Met deze service kan de gebruiker contextueel gelijke termen of zinnen identificeren, een woordenboek opstellen, en daarmee tekst vinden en/of organiseren. Een tekst met verwijzing naar de "The Big Apple" kan hierdoor bijvoorbeeld worden gerelateerd aan "New York City".

Demo <http://concept-expansion-demo.mybluemix.net/>

Input Concept met voorbeeld verwijzingen/synoniemen.

Output Een gerangschikte lijst van termen met een contextuele relatie met het concept.

Het concept "Cities" kan bijvoorbeeld gekozen worden in combinatie met de volgende voorbeeld woorden: "NYC", "DC", "London" en "Big cities". Hierdoor kan de volgende output ontstaan: "New York City", "New York", "Washington DC", "Notting Hill Carnival", "London's", "Pier 94", "Suburbs", "Manhattan", "East Village".

Gebruik

De Concept Expansion service is bedoeld voor verschillende domeinen zoals financiën, rechten, medisch, enz. om een woordenboek van conceptueel gelijke woorden te creëren uit ongestructureerde data. De output kan geëxporteerd worden in verschillende formaten en gevoed worden aan andere Watson services waardoor deze een beter resultaat kunnen leveren.

Colimbra

Deze services zal niet gebruikt worden voor het prototype. Het zou in de toekomst wel gebruik kunnen worden om een verzekeringsgericht woordenboek op te zetten waardoor andere services een beter resultaat kunnen leveren.

3. Concept insights

De Concept Insights service relateert input tekst met een concept graph gebaseerd op de Engelstalige Wikipedia. Deze service identificeert niet alleen expliciet genoemde concepten, maar geeft ook de concepten die niet expliciet zijn genoemd.

Demo <https://concept-insights-demo.mybluemix.net/>

Input Tekst (Unicode) en een collectie van documenten/HTML-pagina's.

Output Gerelateerde documenten (op basis van de vooraf gegeven documenten) en een lijst van genoemde concepten.

Gebruik

Deze service is bedoeld om gebruikers te helpen bij het suggereren van soortgelijke documenten/informatie. Hierdoor kunnen organisaties bijvoorbeeld relevante artikelen, mensen of onderwerpen geven aan een gebruiker.

Colimbra

Deze service wordt niet gebruikt in het prototype. In de toekomst kan deze service gebruikt worden om bijvoorbeeld extra achtergrondinformatie te vinden over verschillende verzekeringen of om een antwoord op een vraag te zoeken via Wikipedia.

4. Dialog

Met de Dialog service kan ontworpen worden hoe de applicatie een conversatie laat verlopen met de uiteindelijke gebruiker. Deze service zorgt ervoor dat de applicatie in natuurlijke taal een reactie kan geven op input van de gebruiker. Hierdoor kunnen bijvoorbeeld vragen beantwoord worden of kan een gebruiker door een proces, applicatie, of moeilijke taak geholpen worden. Deze service is in staat om gebruikersinformatie op te slaan om door te geven aan een backend of om persoonlijkere conversaties te realiseren.

Demo <http://dialog-demo.mybluemix.net/>

Input Conversatie script gebaseerd op expertise over een bepaald domein.

Output De uiteindelijke gebruikers kunnen in natuurlijke taal communiceren met de applicatie via het vooraf gedefinieerde script.

Gebruik

Deze service wordt gebruikt om de gebruikers te assisteren bij het uitvoeren van verschillende taken zoals bijvoorbeeld het resetten van een wachtwoord.

Colimbra

Aangezien door deze service een applicatie in staat is om in natuurlijke taal te communiceren met de gebruiker wordt deze service gebruikt in het prototype. Door deze service kan het antwoord op regelmatig gestelde vragen vooraf gedefinieerd worden.

5. Document Conversion

De Document Conversion service is een eenvoudige conversatie programma zonder cognitieve vaardigheden. Wat deze service interessant maakt is dat documenten geconverteerd kunnen worden naar zogenaamde "answer units" in JSON-formaat. Deze answer units kunnen direct gebruikt worden door andere Watson services zoals de Retrieve and Rank service (paragraaf 10).

Demo <https://document-conversion-demo.mybluemix.net/>

Input PDF, Microsoft Word of HTML document.

Output HTML document, Answer units JSON document of Plain Text Document.

Colimbra

Voor Colimbra is documentconversie niet alleen mogelijk via deze service aangezien Colimbra ook een eigen oplossing heeft voor documentconversie. Voor het prototype zal echter wel gebruik gemaakt worden van deze service omdat deze service het converteren naar "answer units" ondersteund.

6. Language translation

Verzorgt een domein specifieke vertaling van tekst door gebruik te maken van "statistical machine translation" technieken. Deze service ondersteunt op dit moment de volgende talen: "Frans", "Portugees", "Arabisch", "Spaans", "Engels" en "Egyptisch".

Demo <http://language-translation-demo.mybluemix.net/>

Input Tekst in een van de ondersteunde input talen en domeinen.

Output De input tekst vertaald naar de geselecteerde taal.

Colimbra

Deze service zal alleen gebruikt worden door Colimbra wanneer de Nederlandse taal wordt ondersteund. Wanneer de Nederlandse taal ondersteund wordt kunnen Engelstalige bronnen gebruikt worden om een Nederlandstalig antwoord te geven en andersom.

7. Natural Language Classifier

Deze service wordt gebruikt om natuurlijke taal te interpreteren en classificeren. Hiermee kunnen ontwikkelaars zonder een achtergrond in machine learning natuurlijke taal begrijpen binnen hun applicaties. Deze service interpreteert de intentie achter een bepaalde tekst en geeft de bijbehorende classificatie met een zekerheidsscore.

Demo <http://natural-language-classifier-demo.mybluemix.net/>

Input Tekst voor een vooraf getraind model.

Output Classificaties gesorteerd op zekerheidsgraad.

Gebruik

Deze service kan getraind worden voor een specifiek domein of applicatie. Voorbeelden van gebruik zijn:

- Vaak voorkomende vragen filteren om beantwoord te kunnen worden door een applicatie;
- Sms-tekst classificeren als persoonlijk, werk of promotioneel;
- Twitter tweets classificeren als een set van classes zoals evenementen, nieuws of meningen.

Colimbra

Het is de bedoeling dat het prototype volledig wordt gestuurd in natuurlijke taal. Deze service wordt gebruikt om het resultaat van de applicatie te verbeteren door de intentie van een stuk tekst te kunnen achterhalen. Verschillende classificatie instanties kunnen gebruikt worden voor bijvoorbeeld het classificeren of een vraag wel of niet een verzekeringsvraag is en een andere om te achterhalen wat voor antwoord de gebruiker verwacht, vaak ook de "lexical answer type" genoemd.

8. Personality Insights

Wordt gebruikt om een persoon te begrijpen. Met deze service worden persoonlijkheids-karakteristieken en de normen en waarden van een bepaalde persoon bepaald. De Personality Insights service helpt met het achterhalen en analyseren van persoonlijkheidsattributen om een persoonlijke interactie met de gebruiker te kunnen realiseren.

Demo <https://watson-pi-demo.mybluemix.net/>

Input JSON, Tekst of HTML (zoals social media, e-mails, blogs, of andere communicatiebronnen) geschreven door een individu.

Output Een boom van cognitieve en sociale karakteristieken in JSON of CSV-format.

Gebruik

Bedrijven kunnen de gedetailleerde persoonlijkheidsportretten van deze service gebruiken om een persoon beter en/of persoonlijker te kunnen begeleiden. Met deze data is de ontwikkelaar ook in staat om gerichte marketing te tonen en potentieel gewenste producten aan te bieden.

Colimbra

Voor het prototype zal deze service niet worden gebruikt. In de toekomst kan Colimbra deze service inzetten om bijvoorbeeld, bij klanten die daar toestemming voor geven, de social media accounts te doorlopen om met de verzamelde gegevens een persoonlijkheidsportret op te stellen. Dit kan dan weer gebruikt worden om persoonlijker interacties met gebruikers te realiseren.

9. Relationship Extraction

De Relationship Extraction service kan op dit moment alleen gebruikt worden om nieuws artikelen te analyseren. Deze service vindt en labelt de belangrijke onderdelen uit en stuk tekst.

Demo <http://ibmlaser.mybluemix.net/siredemo.html>

Input Tekst afkomstig uit nieuwsartikelen.

Output Een lijst van keywords uit de gegeven tekst en de relaties tussen deze keywords in XML-formaat.

Gebruik

Deze service functioneert het beste op tekst afkomstig uit nieuws artikelen aangezien de service hierop getraind is. Deze service kan bijvoorbeeld gebruikt worden in bedrijven om informatie uit nieuwsartikelen te verzamelen over hun eigen producten en/of de producten van een concurrent.

Colimbra

De Relationship Extraction service wordt niet gebruik in het prototype. Een mogelijk gebruiksscenario is om nieuws afkomstig uit de verzekeringsbranche te verzamelen.

10. Retrieve and rank

De Retrieve and Rank service gebruikt een combinatie van zoek en machine learning algoritmen om "signalen" in data te vinden. Ontwikkelaars uploaden hun data naar deze service, trainen een machine learning model gebaseerd op deze data, en gebruiken daarna dit model om verbeterde zoekresultaten te kunnen bieden aan de uiteindelijke gebruikers.

Demo <http://retrieve-and-rank-demo.mybluemix.net/rnr-demo/dist/#/>

Input Documenten en vragen gerelateerd aan deze documenten.

Output Geïndexeerde documenten, het machine learning model, een lijst van relevante documenten inclusief de metadata.

Gebruik

De Retrieve and Rank service kan ingezet worden in een aantal scenario's waarin informatie opgezocht moet worden. Voorbeelden van gebruik zijn bijvoorbeeld in een helpdesk omgeving voor assistentie bij vraag van een klant of voor een projectmanager om domein experts te vinden voor een projectteam.

Colimbra

Deze service zal gebruikt worden voor het prototype. Deze service zorgt ervoor dat de applicatie door bijvoorbeeld de polisdocumenten kan zoeken voor een antwoord. Ook kan deze service ingezet worden om bijvoorbeeld relevante informatie uit Wikipedia te tonen.

11. Speech to Text & Text to Speech

De speech to text en text to speech services zijn twee soortgelijke services. Deze services gecombineerd zorgen voor de conversatie tussen tekst en audio.

Speech to text:

Demo <https://speech-to-text-demo.mybluemix.net/>

Input Gestreamde audio of opgenomen audio met gesproken taal.

Output Een tekstuele representatie van de gegeven audio.

Text to speech:

Demo <https://text-to-speech-demo.mybluemix.net/>

Input Portugese, Engelse, Franse, Duitse, Italiaanse, Japanse of Spaanse tekst.

Output Audio. De stem van de spreker is afhankelijk van de taal, voor sommige talen kunnen verschillende sprekers beschikbaar zijn.

Colimbra

Deze services zullen in eerste instantie niet gebruikt worden in het prototype. Aangezien de implementatie van deze service waarschijnlijk vrij eenvoudig is kan dit wel toegevoegd worden aan het prototype. Hier ligt echter niet de focus op.

12. Tone Analyzer

De tone analyzer service wordt gebruikt om taal tonen in tekst te achterhalen en te begrijpen. Een taalkundige analyse wordt gebruikt om emotionele en sociale hints te vinden in tekst.

Demo <https://tone-analyzer-demo.mybluemix.net/>

Input Tekst.

Output JSON geformatteerde tekst met een hiërarchische representatie van de analyse.

Gebruik

Een aantal gebruiksscenario's voor deze service zijn:

- Analyse van e-mails, presentaties, of andere klantcommunicatie media;
- Analyse van hoe een lezer een bepaalde blogpost kan interpreteren;
- Helpen met het beter begrijpen van de potentiële impact van woordkeuzes.

Colimbra

Deze service wordt niet gebruikt in het prototype. In de toekomst kan deze service gebruikt worden om bijvoorbeeld te achterhalen waar de meest frustrerende vragen over gaan. Of om, indien nodig, de gebruiker met behulp van een dialoog op zijn gemak te kunnen stellen.

13. Tradeoff Analytics

De tradeoff analytics service helpt mensen met het maken van keuzes wanneer rekening gehouden moet worden met verschillende aspecten. Deze service gebruikt een wiskundige filteringstechniek (Pareto Optimization), waarmee de gebruikers afwegingen kunnen beschouwen tijdens het maken van een keuze.

Demo <http://tradeoff-analytics-demo.mybluemix.net/>

Input Een te maken beslissing met objectieven en opties.

Output JSON geformatteerde objecten waarmee de beste opties worden getoond, inclusief de afwegingen die worden gemaakt.

Gebruik

Deze service kan bijvoorbeeld banken of financiële managers helpen met het analyseren van de beste investeringsstrategie gebaseerd op de risico's en kosten. Gebruikers kunnen ook geholpen worden met de aankoop van een product waarmee bijvoorbeeld de functies, prijs en garantieduur worden vergeleken.

Colimbra

Deze service wordt niet gebruikt in het prototype. In de toekomst kan deze service gebruikt worden om gebruikers te helpen met het kiezen van een verzekering.

14. Visual recognition

Met deze service kan de inhoud van een plaatje of videoframe worden begrepen. Deze service geeft scores voor relevante classificaties waarmee onder andere objecten, evenementen en de omgeving worden getoond.

Demo <https://visual-recognition-demo.mybluemix.net/>

Input Plaatjes in JPEG-formaat.

Output Een set van classificaties inclusief een score van zekerheid.

Colimbra

Deze service wordt niet gebruikt in het prototype. In de toekomst kan deze service gebruikt worden om, door de uiteindelijke klant, geüploade documenten te begrijpen en hier een passende reactie op te geven.

Bijlage E: Prototype

