

Rapportage

Plan van Aanpak

Probleem:

Nu het weer bijna zomer wordt zijn veel Families en vrienden weer buiten. Echter zit het weer nu in de lente niet altijd mee. Vaak gaan mensen naar buiten en worden ze getroffen door regen of een bewolkte dag. De kans is dat dit ook zo doorgaat tijdens de zomer. Mijn opdrachtgever zag dit als een probleem en zei, "Wat deze mensen nodig hebben is een AI die ze het weer kan tonen en advies kan geven of het een handige zet is om een buitenactiviteit uit te voeren". Dit is dus wat ik ga maken.

Algoritme:

Het gekozen algoritme voor de AI zal gebruik maken van basis NLP (natural language processing) en een weather API. NLP implementeer een algoritme om de tekst die het binnenkrijgt te onderzoeken en de tekst weer op te bouwen in een manier dat de AI het kan begrijpen. Dit doet hij bijvoorbeeld door nutteloze context zoals hier, daar, de, het ,een, enz weg te halen en meer te focussen op woorden die echt een betekenis hebben. Dit is het deel zinsontleding van het algoritme. Het weer en advies deel doet ie met behulp van een weather API. De AI kijkt naar de context van de zin en op basis daarvan kan die kijken naar het de weather API. Er zal een json bestand komen met een hoop buitenactiviteiten om zo te kijken of de huidige temperatuur buiten goed is voor de activiteit. Als het geen goed idee is om de activiteit uit te voeren zal de AI een alternatief geven aan de gebruiker.

Planning:

Taken:

Maak verbinding met weather api

Maak json file met activiteiten

Maak een frontend waar de vragen kunnen worden gesteld

Maak het algoritme voor het geven van advies

Maak het algoritme die kijkt of er advies gegeven kan worden met het json bestand

Maak het algoritme voor de zinsontleding

Rapportage, documentatie, poster

Extra uitstel

	Ma	Di	Wo	Do	Vr	Za	Zo
D8							
D9							
D10							

Ik denk dat ik alle tijd kan gebruiken. Kans is groot dat er extra's zullen zijn die ik wel wil toevoegen, maar die teveel tijd zal kosten.

Risico's:

Risico (matig groot):

Er kunnen activiteiten worden gevraagd die niet in mijn json bestand staan.

Oplossing(en):

Een vast alternatief als dat het geval blijkt te zijn. Zoals het weer opgeven om de gebruiker zelf te laten bepalen.

Risico (groot):

De code kan de context van de tekst verkeerd begrijpen

Oplossing(en):

Kijken of een bepaalde context daadwerkelijk een bijbehorend advies zou kunnen opleveren door bijvoorbeeld unittests te maken om dit soort problemen te voorkomen.

Risico (groot):

Het algoritme voor zinsontleding kan moeilijker zijn dan verwacht en zal dus extra tijd kunnen kosten.

Oplossing(en):

- Aangezien dit nieuw is meer voorstudie doen voordat de ipass begint als goede voorbereiding
- meer werken in versie. Een functionele versie maken die nog niet optimaal is, maar die wel wordt bewaard mochten andere versie niet meer werken.

Terugblik

Wat was anders:

In mijn vorige PvA was mijn planning een stuk langer. In plaats van 5 weken bleken het maar 3 weken te zijn. Dit betekende dat ik mijn planning een stuk moest inkorten. Ook was vergeten de tijd voor de rapportage, documentatie en poster in te plannen. De planning die boven staat is de nieuwe planning die ik na de feedback had gemaakt. Het algoritme en de volgorde waarin het algoritme is gemaakt is hetzelfde gebleven.

De frontend zou eigenlijk met tkinter gemaakt worden. Alleen vind ik tkinter niet leuk. Dus heb ik tutorials opgezocht over flask en heb besloten dat te gebruiken.

Ik heb wel iets gemaakt wat ik eerst niet wou implementeren, maar wel noodzakelijk was voor mij en dat was een deel hardcoded checks. Wat ik hiermee bedoel is dingen zoals de manieren waarop je voor een datum kan vragen (morgen, dinsdag, week, enz) worden opgeslagen in lijsten en worden vergeleken met de zinscontext. Dit heb ik gedaan omdat dit de snelste oplossing was en ik nog te weinig kennis heb over NLTK en dates om dit proces te maken zonder deze waardes.

Beslissingen:

De enige beslissing die anders is geworden was de manieren waarop data werd opgeslagen in het algoritme zelf. Ik was van plan data in lists te bewaren aangezien lists makkelijk op te halen, scannen en te veranderen zijn. Echter kwam ik er snel achter dat ik bij een functie geen dubbele data mag terugsturen en bij andere functies dat ik nog waardes moest koppelen aan keywords. Alleen lists gebruiken was al snel geen goed idee meer.

De json file is zoals ik had gedacht. Een simpel bestand met een hoop activiteiten die een min en max temperatuur hebben. De temperaturen zijn vaste waardes en niet random waardes (zoals ik er eerst over dacht). Dit is omdat random waardes me ontzettend onlogisch leek. Als het de ene dag 20 graden is en je kan dan wel gaan basketballen, zou het onlogisch zijn als het weer eens 20 graden is en dan opeens is basketballen niet meer handig. Als een activiteit niet in de lijst staat stuurt de AI de gevraagde temperatuur om de gebruiker zelf de keuze te laten maken.

Bronnen:

Flask:

- Tech With Tim. (2020, 24 april). Flask tutorials [Videobestand]. Geraadpleegd van https://www.youtube.com/playlist?list=PLzMbGfZo4-n4vJJyUVV3Un_NFS5EOgX

- *How to add more than one item to the session in flask.* (2016, 6 januari). Geraadpleegd op 20 juni 2020, van <https://stackoverflow.com/questions/34630709/how-to-add-more-than-item-to-the-session-in-flask>
- *Sending data from HTML form to a Python script in Flask.* (2012, 19 juli). Geraadpleegd op 18 juni 2020, van <https://stackoverflow.com/questions/11556958/sending-data-from-html-form-to-a-python-script-in-flask>

Nltk:

- Joe James. (2019, 28 februari). Python NLTK [Videobestand]. Geraadpleegd van https://www.youtube.com/watch?v=gRk53jBPYvE&list=PLj8W7XlvO93pDdmGpOlivqSwXFU_h4e2Z

Pytest:

- *Installation and getting started.* (z.d.). Geraadpleegd op 22 juni 2020, van <https://docs.pytest.org/en/latest/getting-started.html#getstarted>

Pdoc:

- *Package pdoc.* (z.d.). Geraadpleegd op 24 juni 2020, van <https://pdoc3.github.io/pdoc/doc/pdoc/#gsc.tab=0>

Python:

- *How to retrieve an element from a set without removing it.* (2008, 12 september). Geraadpleegd op 18 juni 2020, van <https://stackoverflow.com/questions/59825/how-to-retrieve-an-element-from-a-set-without-removing-it>
- *Python Dictionaries.* (z.d.-b). Geraadpleegd op 19 juni 2020, van https://www.w3schools.com/python/python_dictionaries.asp

Api:

- *Current weather data.* (z.d.). Geraadpleegd op 6 juni 2020, van <https://openweathermap.org/current>
- *5 day weather forecast.* (z.d.). Geraadpleegd op 10 juni 2020, van <https://openweathermap.org/forecast5>