Reinforcement Learning voor het classificeren van werkkosten

Haroun Payandeh

Maart 2020

Reinforcement Learning voor het classificeren van werkkosten

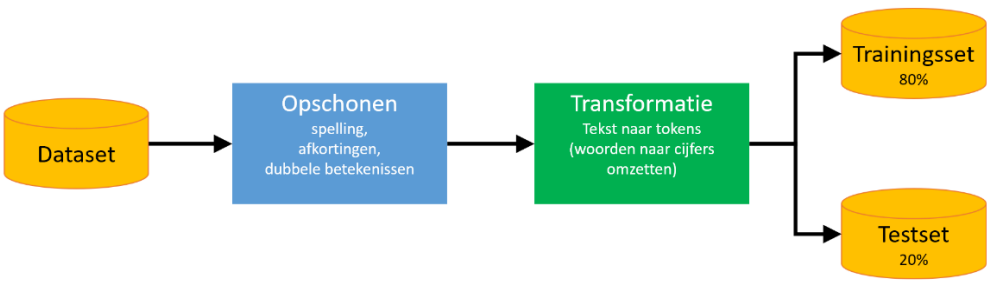
Dit artikel is een vervolg op een eerder artikel genaamd 'De werkkostenregeling controleren met een computer, kan dat?'

Inleiding

Het is al meer dan drie jaar geleden dat werkkostenregeling (WKR) verplicht werd gesteld voor alle organisaties. De regeling houdt in dat 1.2% van de fiscale loonsom door werkgevers onbelast vergoed mag worden. Hierdoor is er al heel snel behoefte ontstaan voor een (intelligent) systeem dat de medewerkers van de organisatie kan helpen bij het verzamelen en indelen van kosten die gerelateerd zijn aan de WKR. De boekingsomschrijving in de administratie kan gebruikt worden voor het opzetten van een Deep Learning Model. In deze blog zal ik het reinforcement learning principe gebruiken voor het vinden van WKR-gerelateerde kosten en het classificeren van deze kosten voor de WKR op basis van de boekingsomschrijving. Deze toepassing past binnen het werkveld Natural Language Processing, ofwel NLP.

Opschonen dataset

Voordat het model wordt opgezet is het van belang om de beschikbare dataset te analyseren. Met andere woorden, wat zijn de kenmerken van de dataset waar we rekening mee dienen te houden? De dataset die we gebruikt hebben voor deze blog bestaat uit 150.000 boekingsregels uit een groot aantal verschillende administraties. Elke boekingsregel bevat een unieke boekingsomschrijvingen en een bijbehorend WKR-label. De gebruikte WKR-labels geven aan of een kostenpost WKR-gerelateerd is of niet, mogelijk WKR-gerelateerd is, dan wel niet te classiferen is. De categorieën zijn niet in balans, ongeveer 90 procent van de omschrijvingen behoren tot de groep 'niet-WKR'. Verder bevatten de omschrijvingen veel afkortingen en spelfouten, ook zijn ze vaak erg kort, slechts 3 à 4 woorden. Doordat de boekingsomschrijvingen zo kort zijn bevatten ze weinig contextuele informatie. Dit gegeven in relatie met de 'normale' NLP-problemen zoals spelfouten, afkortingen, woorden met dubbele betekenissen, vormt wat deze dataset betreft een enorme uitdaging. We zullen de dataset daarom eerst zo goed mogelijk moeten opschonen, dat wil zeggen het verwijderen van spelfouten en lidwoorden en het corrigeren van afkortingen. Verder zullen we een oplossing moeten zien te vinden voor woorden met dubbele betekenissen. De opgeschoonde dataset zal als trainingset voor het model worden gebruikt. Hierna is het voorgaande proces in een schema uitgwerkt:



Toelichting:

- **Spelfouten** Om spelfouten te corrigeren, gebruik ik een lijst met woorden die frequent worden gebruikt binnen het onderzoeksdomein. De lijst omvat niet alleen woorden, maar ook informatie over de waarschijnlijkheid van het voorkomen van een woord, gegeven het onderzoeksdomein.
- **Afkortingen** Afgekorte woorden worden volledig uitgeschreven. Ook hier gebruiken we een lijst die is afgestemd op het onderzoeksdomein.
- **Dubbele betekenissen** Om te kunnen omgaan met onbekende woorden die een relatie hebben met woorden in de trainingsset, moet een oplossing worden gezocht. Stel dat de trainingsset het woord 'appel' bevat, hoe kun je het model dan leren om het woord 'peer' te herkennen? Dit probleem lossen we op door de woorden te transformeren naar zogeheten 'woordvectoren', ofwel een geometrische weergave van woorden. Met deze vectoren kun je relaties tussen woorden in kaart brengen, zodat het model het woord 'peer' zal herkennen als een woord dat een sterke relatie heeft met het woord 'appel'. Mooi is dat er zogenaamde 'dictionaries' met woordvectoren beschikbaar zijn die andere onderzoekers al hebben opgebouwd, waarvan je gebruik kunt maken. Dit heeft zelfs de voorkeur boven het zélf opbouwen van zo'n dictionary, omdat de kwaliteit van een dictionary toeneemt als deze afkomstig is van modellen die zijn getraind op enorm grote datasets.

Deep Learning Model

Het is nu tijd om het model te kiezen dat we gaan gebruiken voor het uitvoeren van de classificatie van de boekingsomschrijvingen. We hebben een model nodig dat onderscheid kan maken tussen kernwoorden en minder belangrijke woorden in de omschrijving. Hierna een voorbeeld:

'*Telefoon abonnement medewerker Piet Jansen*'

Het model moet nu de woorden 'abonnement' en 'telefoon' een hoger gewicht toekennen dan de overige woorden, gegeven het feit dat we zoeken naar WKR-gerelateerde kosten. Verder moet het model om kunnen gaan met de woordvolgorde. Het mag namelijk niet uitmaken of in het hiervoor genoemde voorbeeld de omschrijving als volgt luidt:

'*Piet Jansen abonnement telefonie*'

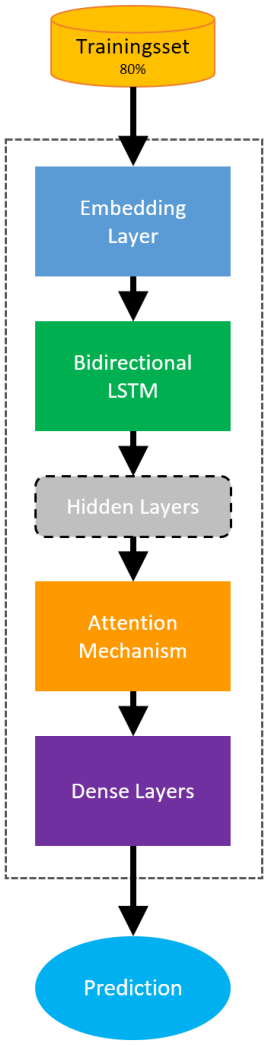
Een ander aspect is dat het model moet kunnen herkennen dat woorden in een andere context worden gebruikt en daardoor een omschrijving juist wel of juist niet voorziet van het label 'WKR-relevant'. Zo is de volgende boekingsomschrijving niet relevant voor de WKR:

'*Abonnement telefonie KPN Lokale Overheid*'

We kiezen – na wat experimenteren – voor een 'Bidirectional LSTM-model'. Dit is een Deep Learning model dat teksten van twee kanten benadert (vandaar de naam 'bidirectional'). Daardoor is dit model vooral geschikt voor probleemstellingen waar concepten in verschillende volgordes in teksten voorkomen. Een aanvulling is dat we een 'Attention Mechanisme' toevoegen dat door Google is beschreven in de paper

'*Attention is all you need*'. Met dit mechanisme kunnen we woorden in een tekst onderling tegen elkaar afwegen en van een gewicht voorzien.

De technische architectuur van het model ziet nu als volgt uit:



De embedding layer bestaat uit een matrix met dimensie (N x 300), waarbij N gelijk is aan het aantal woorden in een domeinspecifiek vocabulaire, bestaande uit woorden die voorkomen in een diverse administraties. In deze matrix wordt ieder woord uit het vocabulaire omgezet naar een woordvector met behulp van de eerder genoemde dictionaries afkomstig uit de data-science gemeenschap.

Vanuit de Embedding Layer worden de in vectoren omgezette woorden doorgegeven aan het Bidirectional LSTM, dat vervolgens de Hidden Layers van het Deep Learning model aanmaakt.

De Hidden Layers worden vervolgens door het Attention Mechanisme gebruikt om de gewichten van woorden in een zin te bepalen. Als laatste gebruiken we lineaire Dense Layers om de classificatie uit te voeren.

Het resultaat

Na het trainen van het model volgt uiteraard het uitvoeren van een test op een dataset die het model nog niet eerder heeft gezien. In ons geval is dat de testset van 20%, die is afgesplitst van de set met trainingdata. Het model zoals hiervoor beschreven realiseert een precisie van 0.91 en een recall van 0.87. Precision en recall zijn meetwaarden die gebruikt worden om de kwaliteit van het model te bepalen. In een eerder artikel op onze website wordt uitgelegd wat deze meetwaarden betekenen en hoe ze worden gebruikt.

Over de behaalde resultaten zijn we uitermate enthousiast. Het model weet bijna 9 van de 10 keer een boekingsomschrijving correct te classificeren. Dit percentage ligt aanmerkelijk hoger dan het tot nu toe gehanteerde model dat werkt met een 600-tal steekwoorden.

Verbeterpunten

Vervolgens hebben we gekeken naar waar we het model nog zouden kunnen verbeteren. Bij de analyse van de onjuist gelabelde boekingsomschrijvingen zagen we dat de oorzaak bijna altijd was gelegen in omschrijvingen die ‘niets zeggend’ waren. Een voorbeeld hiervan is bijvoorbeeld de omschrijving ‘abonnement 2019’. Geen enkel model zal hier een betere labeling tot stand kunnen brengen.

Wat we zagen is dat het model wel verbeterd zou kunnen worden door het toevoegen van extra ‘features’. Dit zijn aanvullende data-elementen die extra informatie verschaffen over de WKR-classificatie. Voorbeelden zijn de achterliggende crediteur en de omvang van de bedragen.

We hebben uiteindelijk een zevental factoren kunnen onderkennen die kunnen worden ingezet om het model te verbeteren. Deze hebben we toegevoegd aan de trainingset, waarna we het model opnieuw hebben getraind. De nieuwe waarden voor precisie en recall zijn respectievelijk 0.92 en 0.9 (was 0.91 en 0.87) procent. Het toevoegen van de aanvullende factoren leidt dus vooral tot een verbetering van de recall.

Conclusie

We hebben het ontwikkelde model operationeel gemaakt in ons data-analytics platform DBI. Daarbij hebben we het ‘oude’ model dat op steekwoorden en fuzzy zoeken is gebaseerd niet weggegooid. Sterker nog, we hebben het Deep Learning Model gecombineerd met het Steekwoorden Model. Daardoor is een hybride model ontstaan dat de voordelen van beide modellen weet te benutten.

Meer weten?

Wil je meer weten over het toepassen van Machine Learning en/of Deep Learning? Wil je zien hoe het werkt op je eigen dataset? Wil je ondersteuning krijgen bij het concreet opzetten en in productie nemen van dergelijke modellen? Schroom niet om contact op te nemen via h.payandeh@duroi.nl.