# Bachelorscriptie

| | |
|---|---|
| **Naam student** | Steven Veerbeek |
| **Begeleider** | Prof. dr. Yoad Vinter Seggev |
| **Tweede beoordelaar** | Drs. Frans Adriaans |
| **Opleiding** | Bachelor Kunstmatige Intelligentie, Universiteit Utrecht |
| **ECTS** | 7,5 |

# Automatic image captioning

## Methodes en modellen beschreven

## 1. Inleiding

Hoe leer je een computer om een foto te beschrijven in natuurlijke taal? Eén van de vereisten is dat hij objecten kan herkennen, zoals bijvoorbeeld auto's. Om een computer dit te leren is het echter niet voldoende om enkel een specificatie van de vorm en kleur van een auto te geven. Auto's komen immers in verschillende vormen, maten en kleuren, en zien er bovendien vanuit elk perspectief weer anders uit. En zelfs al ís een computer zo ver dat hij een auto kan herkennen, dan nog is dit niet genoeg om een complete scène te beschrijven in een goed lopende zin. Dit is namelijk een taak waarbij niet alleen de losse objecten herkend moeten worden, maar waarbij ook de onderlinge relaties tussen objecten van groot belang zijn. Kortom: het laten beschrijven van een foto door een computer is geen eenvoudige taak en is niet één-twee-drie op een intuïtieve manier op te lossen.

De taak van het genereren van beschrijvingen bij foto's, die ik voortaan zal aanduiden als *automatic image captioning* (AIC), heeft het afgelopen decennium veel aandacht gekregen binnen het gebied van de kunstmatige intelligentie, en specifiek in het veld van de natuurlijke taalverwerking en computer vision. Deze scriptie gaat over de *state-of-the-art* AIC-methodes. Aan de hand van literatuuronderzoek geef ik antwoord op de vraag wat in de afgelopen jaren de meest gebruikte methodes en modellen zijn binnen AIC en hoe deze presteren.

Kortgezegd is de achterliggende gedachte van AIC om een computer te leren om beelden te beschrijven op een manier die doet denken aan de manier waarop een kind dat leert: door herhaling. Een kind wordt bijvoorbeeld herhaaldelijk blootgesteld aan beelden van auto's in combinatie met het horen van het woord 'auto'. Daaruit kan het kind vervolgens een universeel begrip van het woord vormen. Ook leert het door talloze herhalingen hoe het een welgevormde zin kan produceren waarmee het complexere taferelen kan beschrijven. AIC-systemen maken vaak gebruik van kunstmatige neurale netwerken die, op vergelijkbare wijze werken. Aan de hand van duizenden trainingsvoorbeelden worden deze netwerken getraind om bij een inputafbeelding bijpassende beschrijvingen in natuurlijke taal te genereren. In tegenstelling tot de grote hoeveelheid tijd en geld die het kost om afbeeldingen door mensen te laten annoteren, maken deze systemen het mogelijk om op zeer snelle en efficiënte wijze grote verzamelingen afbeeldingen van beschrijvingen te voorzien.

Hoewel het gebied van AIC nog erg jong is, bouwt het voort op een basis die al eerder is gelegd. Rond de millenniumwisseling werden de eerste systemen gebouwd die aan de hand van visuele informatie trefwoorden konden produceren bij beelden (bijv.: Mori et al., 1999, Duygulu et al., 2002). AIC steunt voor een groot deel op dit concept, maar voegt hier nog een cruciale stap aan toe: zodra de losse trefwoorden eenmaal zijn gegenereerd, worden hier zinnen van gevormd. Daarvoor moet niet

alleen herkend worden welke objecten en attributen in een foto zichtbaar zijn, maar ook hoe deze tot elkaar in relatie staan. Bovendien moeten deze relaties in natuurlijke taal worden uitgedrukt, wat betekent dat er een taalmodel nodig is.

Een AIC-systeem bestaat uit verschillende componenten die elk een deel van dit proces uitvoeren. Hoewel de structuur en werking van deze componenten sterk kunnen verschillen van systeem tot systeem, volgen de meeste systemen een globaal stappenplan dat min of meer overeenkomt. Voordat ik hier in de komende hoofdstukken dieper op in ga, geef ik in het kort een overzicht van de rest van deze scriptie.

In hoofdstuk 2 behandel ik de eerste stap in het stappenplan: visuele detectie. Uit een inputafbeelding moeten eerst de losse objecten, attributen en relaties ontdekt en benoemd worden. Hiervoor bestaan verschillende strategieën en modellen, die ik in het betreffende hoofdstuk zal behandelen.

Vervolgens gaat hoofdstuk 3 over de tweede stap: taalproductie. Op basis van de gedetecteerde woorden moeten beschrijvende zinnen gevormd worden. Ook dit probleem wordt binnen de wetenschappelijke literatuur op verschillende manieren benaderd, variërend in nauwkeurigheid en originaliteit van de gegenereerde beschrijvingen.

In hoofdstuk 4 bespreek ik de prestaties van de modellen die ik in hoofdstuk 2 en 3 heb behandeld. Dit doe ik aan de hand van de testresultaten die in de papers worden vermeld. Ook probeer ik, waar mogelijk, de modellen met elkaar te vergelijken.

In de discussie in hoofdstuk 5 wil ik het kort hebben over het toekomstperspectief van AIC en deel ik een aantal ideeën over mogelijke nieuwe toepassingen en verbeteringen binnen het gebied.

# 2. Visuele detectie

In dit hoofdstuk zal ik dieper ingaan op de eerste stap in het image captioning-proces. Voordat het AIC-systeem een beschrijving kan genereren, moet bekend zijn wat er op de foto te zien is. Zoals wij mensen objecten, personen, en relaties tussen deze entiteiten kunnen herkennen door naar een foto te kijken, moet ook een AIC-systeem deze informatie op de een of andere manier kunnen extraheren uit de visuele informatie van een foto. Bij de meeste systemen begint het proces dan ook met een detectiestap waarin objecten, attributen en soms ook relaties daartussen worden gedetecteerd. Dit gebeurt door middel van een woorddetector die wordt getraind om deze dingen in een foto te herkennen en benoemen. De output van zo'n woorddetector is vaak een reeks woorden die corresponderen met de gedetecteerd objecten of attributen in de foto.

Een veelgebruikte basis voor deze woorddetectors is een Convolutional Neural Network (CNN). In de volgende paragraaf leg ik de globale werking van een CNN-woorddetector uit aan de hand van een voorbeeld dat gebaseerd is op Krizhevsky et al. (2012), en in paragraaf 2.2 beschrijf ik een aantal verschillende implementaties hiervan.

### 2.1 Werking van een CNN-woorddetector

De inputlaag van een CNN-woorddetetcor ontvangt de (eventueel gedownscalede) pixels van de foto, waarna een aantal verborgen lagen volgen. Elk van de neuronen in de outputlaag representeert een woord. Dit kunnen woorden van verschillende woordsoorten zijn, zoals zelfstandige naamwoorden (bijvoorbeeld objecten), werkwoorden (acties/relaties) of bijvoeglijke naamwoorden (attributen). Deze verzameling woorden wordt vooraf vastgesteld en toegewezen aan de outputneuronen. Vervolgens wordt het netwerk aan de hand van een

grote verzameling gelabelde afbeeldingen getraind om deze woorden te herkennen en via de outputlaag aan te geven welke woorden in de foto zijn gedetecteerd. Door aan meerdere instanties te worden blootgesteld van afbeeldingen die gelabeld zijn met hetzelfde woord, updatet het netwerk zijn 'kennis' van dit woord en leert hij dit woord te herkennen in nieuwe afbeeldingen. Hoe hoger de activatiewaarde van een outputneuron, hoe 'zekerder' de detector ervan is dat de referent van het bijbehorende woord zichtbaar is in de foto. Deze activatiewaarden zal ik voortaan aanduiden als 'zekerheidswaarden'.

De specifieke implementatie en trainingswijze van de woorddetector verschilt per systeem. Zo kan er gevarieerd worden in het aantal verborgen lagen en de grootte van de input- en outputlagen. Ook is het feit dat de detector getraind wordt op een beperkte, vaststaande verzameling woorden bepalend voor welke objecten en attributen herkend kunnen worden en hoe deze benoemd worden. Het hangt namelijk af van deze verzameling hoe specifiek de detector is in het benoemen van objecten. Een detector die getraind is op een zeer algemene verzameling van objectklassen zal bijvoorbeeld een vogel kunnen herkennen, maar een detector die speciaal op vogels is getraind zal wellicht ook herkennen dat het om een groene specht gaat. Vaak worden woorddetectors binnen AIC getraind op de meest voorkomende woorden in de labels van de trainingsset. In zo'n geval is dus de inhoud van de trainingsset bepalend voor welke woorden de detector kan herkennen.

### 2.2 Implementaties van CNN-woorddetectors

Fang et al. (2015) trainen twee objectdetectoren die elk gebaseerd zijn op een andere CNN: de een op de CNN van Krizhevsky et al. (2012, AlexNet) en de andere op de 16-lagige CNN van Simonyan & Zisserman (2014, VGG). In de testfase, die ik in hoofdstuk 5 beschrijf, worden de resultaten van beide CNN's vergeleken om te bepalen welke het beste presteert. De netwerken werden getraind door *Multiple Instance Learning* (MIL) toe te passen. Dit is een zgn. 'weakly supervised' methode die het mogelijk maakt om een woorddetector te trainen zonder de noodzaak om alle objecten in de foto's van de trainingsdata nauwkeurig te labelen. De trainingsset bestaat slechts uit een verzameling afbeeldingen met bijbehorende bijschriften. Hoewel dus niet is aangegeven wáár een bepaald object zich in een afbeelding bevindt, kunnen we aan de hand van deze bijschriften wel weten óf het zich in de afbeelding bevindt. Het te leren vocabulaire *V* bestaat uit de 1000 meest voorkomende woorden in de trainingsbijschriften. De output van de woorddetector bestaat uit de verzameling $\tilde{V} \subset V$ van woorden die met de grootste precisie worden gedetecteerd.

Elliot & De Vries (2015) werken met een voorgetrainde *Regions with Convolutional Neural Network features object detector* (R-CNN) van Girshick et al. (2014). Deze objectdetector geeft niet alleen een verzameling woorden als output, maar ook de bijbehorende kaders waarbinnen de gedetecteerde objecten zich bevinden. Waar de meeste AIC-systemen slechts gebruik maken van de gedetecteerde woorden om zinnen te vormen, doet het systeem van Elliot & De Vries nog een extra detectiestap om de nauwkeurigheid van de resulterende zinnen te vergroten. Na het detecteren van de objecten en de bijbehorende kaders, brengt het systeem de onderlinge spatiële relaties tussen de objecten in kaart door middel van een Visual Dependency Representation (VDR). Afhankelijk van hoe de objectkaders zich tot elkaar verhouden in de foto, bepaalt de VDR-parser de spatiële relatie tussen objectparen ('beside', 'above', 'below', 'on', of 'surrounds') en verwerkt hij deze in een boomstructuur. Elke knoop van deze boom stelt een gedetecteerd object voor en elke verbinding

tussen een ouder- en kindknoop stelt een relatie voor tussen deze objecten. In paragraaf 3.2 leg ik uit hoe deze aanvullende informatie wordt geïncorporeerd in de te vormen beschrijving.

Karpathy & Fei-Fei (2015) gebruiken net als Elliot & De Vries de R-CNN van Girshick et al. als basis voor hun objectdetector. Echter, in plaats van de verzameling gedetecteerde woorden te gebruiken die het netwerk als output geeft, representeren ze een afbeelding als een verzameling vectoren die worden berekend aan de hand van de activaties van de laatste hidden layer van de CNN. Elk van deze vectoren representeert weer een regio binnen de foto die is gedetecteerd door het netwerk. Met behulp van een Bidirectional Recurrent Neural Network (BRNN) worden ook de woorden uit de trainingszinnen gerepresenteerd als vectoren van dezelfde dimensie. Door op deze manier de woorden en fotoregio's te projecteren in een gezamenlijke, multimodale ruimte is het mogelijk om woorden en delen van foto's direct met elkaar te vergelijken. Een 'alignment objective' geeft aan de hand van een score aan in hoeverre en fotoregio en een woord—of een volledige foto en een zin—met elkaar corresponderen. Deze score wordt later in de generatiestap gebruikt om beschrijvingen te genereren.

# 3. Taalproductie

Dit hoofdstuk is gewijd aan de tweede en laatste stap in het image captioning-proces: het verwerken van de gedetecteerde woorden tot beschrijvende zinnen. Daarbij maak ik onderscheid tussen twee verschillende strategieën om dit doel te bereiken (naar Bernardi et al., 2016). Bij *description retrieval* wordt een beschrijving gebouwd op basis van beschrijvingen van afbeeldingen uit de trainingsdata die visueel vergelijkbaar zijn met de inputafbeelding. In plaats van een nieuwe zin te genereren, worden hier dus zinnen in zijn geheel overgenomen of samengesteld uit delen van bestaande zinnen. In het geval van *description generation* worden direct op basis van de visuele input nieuwe beschrijvingen gegenereerd door middel van een complexer taalmodel. In dit hoofdstuk ga ik dieper in op de taalmodellen binnen deze laatste categorie. Per paragraaf beschrijf ik één type een geef ik voorbeelden van implementaties. Bij het laatste taalmodel, dat ik in paragraaf 3.3 behandel, zal ik een kleine 'technical case study' houden door een implementatie ervan uit te lichten en deze uitvoerig te beschrijven.

### 3.1 Recurrent Neural Network

Recurrent Neural Networks (RNN's) zijn in de afgelopen jaren uitgegroeid tot veelgebruikte basis voor taalmodellen. Mikolov et al. (2010) bouwden een simpel RNN-taalmodel dat woord voor woord een zin genereert, waarbij elk volgende woord wordt bepaald aan de hand van een kansverdeling die afhangt van de context. Deze context bestaat uit een hidden layer die de willekeurig lange reeks van tot dusver gegenereerde woorden representeert. Dit betekent dus ook dat het netwerk in staat moet zijn om informatie voor een willekeurig lange tijd vast te houden. Voor feedforward netwerken vormt dit een probleem, aangezien deze maar een zeer beperkt 'geheugen' hebben. RNN's daarentegen staan toe dat informatie terugvloeit en daardoor binnen het netwerk kan blijven circuleren, waardoor deze langer 'onthouden' kan worden. Dit maakt RNN's een uitermate geschikte basis voor een taalmodel.

Het Multimodale RNN (MRNN) taalmodel van Karpathy & Fei-Fei (2015) is in feite een uitbreiding van het basismodel van Mikolov et al. Dit model maakt het mogelijk om het generatieproces te conditioneren op een inputafbeelding. Dankzij de projectie van beeld en

tekst in een multimodale ruimte (zie hoofdstuk 2) is het mogelijk om bij het genereren van een zin niet alleen de voorafgaande woorden als context te gebruiken, maar ook de representatie van de inputafbeelding. Bij de eerste iteratie wordt het eerste woord geïnitialiseerd met een speciale START-vector en worden de hidden states berekend op basis van de afbeeldingsvector. Hiermee is de visuele informatie uit de afbeelding opgenomen in de context, en zullen de kansverdelingen voor nieuwe woorden hier voortaan door worden beïnvloed. Vervolgens wordt op basis van het laatst gegenereerde woord (de START-vector in dit geval) en de hidden states het eerste woord gegenereerd. Dit proces herhaalt zich totdat het END-teken wordt gegenereerd en de zin af is. In het paper noemen de auteurs dat het gebruik van *beam search* (zie paragraaf 3.3) de resultaten nog kan verbeteren.

### 3.2 Template-based taalmodel

Template-based taalmodellen zijn relatief simpele modellen die hun zinnen opbouwen volgens een standaard sjabloon. In dit sjabloon hoeven alleen nog een aantal woorden ingevuld te worden op de lege plaatsen. Vanwege dit sjabloon zijn deze taalmodellen niet flexibel in het soort zinnen dat ze genereren: alle zinnen krijgen dezelfde syntactische structuur.

In de boomstructuur die de VDR-parser (zie hoofdstuk 2) van Elliot & De Vries (2015) opbouwt, verbindt elke spatiële relatie twee objecten met elkaar: een *head* (ouderknoop) en een *child* (kindknoop). Deze twee objecten zullen in de te genereren beschrijving respectievelijk fungeren als onderwerp en lijdend voorwerp. Voor elke combinatie van head, child en bijbehorende relatie in de VDR wordt, op basis van de statistieken van de zinnen in de trainingsset, bepaald welk werkwoord het meest waarschijnlijk is om deze relatie uit te drukken. Zodra aan alle relaties in de VDR-structuur een werkwoord is toegewezen, wordt de beschrijving opgebouwd volgens het volgende sjabloon:

DT **head** is V DT **child**.

Hierbij is DT een lidwoord, V het werkwoord, en **head** en **child** de labels van de objecten die onderdeel zijn van de relatie. Het model produceert dus hoofdzakelijk zinnen van het type ‘A man is using a laptop.’ Als de VDR-parser geen relaties tussen objecten kan herkennen, wordt een zin gegenereerd volgens het sjabloon ‘A/An **object** is in the image’, waarbij **object** wordt vervangen door het object met de hoogste zekerheidswaarde (zie paragraaf 2.1). Aan elke kandidaatbeschrijving wordt een score toegekend, gebaseerd op de zekerheidswaarden van de objecten en de statistieken van de zinnen in de trainingsset. De zin met de hoogste uiteindelijke score is de ‘winnaar’ en wordt gekozen als output.

### 3.3 Maximum entropy taalmodel

In deze laatste paragraaf beschrijf ik het maximum entropy taalmodel. Ten opzichte van de voorbeelden die ik in de vorige paragrafen heb gegeven, zal ik in deze paragraaf bij wijze van een ‘technical case study’ een uitgebreidere specificatie van een implementatie geven.

Het maximum entropy principe werd door Berger et al. (1996) als volgt omschreven: ‘Intuitively, the principle is simple: model all that is known and assume nothing about that which is unknown. In other words, given a collection of facts, choose a model which is consistent with all the facts, but otherwise as uniform as possible.’ Vervolgens beschrijven de auteurs een model dat voor een willekeurig proces, gegeven een context *x*, een kansverdeling geeft voor de output *y*. In het geval van een AIC-taalmodel is de output telkens het volgende woord in de gegenereerde zin, en bestaat de

context uit een combinatie van de reeks voorgaande woorden en de gedetecteerde informatie uit de inputafbeelding.

*Technical case study: het taalmodel van Fang et al. (2015)*

Een recent voorbeeld van een toepassing van een ME-taalmodel is het paper van Fang et al. (2015). Hun statistische taalmodel bouwt de zin woord voor woord op en maakt hierbij voor elk volgende woord $w_l$ een kansverdeling over alle woorden in V op basis van een context die uit twee componenten bestaat: de reeks van voorgaande woorden $w_1, w_2, \ldots, w_{l-1}$ en een lijst van gedetecteerde woorden met hoge zekerheidswaarden $\tilde{V}_l \subset \tilde{V}$ (zie paragraaf 2.2 voor een beschrijving van de woorddetector). Het conditioneren van het proces op deze lijst woorden helpt om zoveel mogelijk gedetecteerde woorden te verwerken in de zin. Zodra één van deze woorden wordt opgenomen in de zin, wordt het verwijderd uit de lijst om te voorkomen dat woorden meerdere keren worden gebruikt.

De kansverdeling over de mogelijke volgende woorden wordt steeds berekend met een vijftal 'max-entropy features'. Deze features nemen het kandidaatwoord en de inhoud van de context als argumenten en geven als output een booleanwaarde (0 of 1) of een reëel getal tussen de 0 en 1. Aan elk van deze features wordt een gewicht toegekend dat bij het berekenen van de kansverdeling wordt vermenigvuldigd met de waarde van de feature en zo bepaalt hoe zwaar de betreffende feature meetelt. Voorbeelden van features die meetellen zijn het al dan niet aanwezig zijn van het kandidaatwoord in de lijst met gedetecteerde woorden (0 of 1 als waarde) en, indien dit het geval is, de log-waarschijnlijkheid van het kandidaatwoord zoals bepaald door de woorddetector (reëel getal tussen 0 en 1 als waarde).

Bij het trainen van het taalmodel wordt de log-waarschijnlijkheid van de zinnen in de trainingsset als doelfunctie gesteld. Deze wordt berekend aan de hand van de log-waarschijnlijkheden van de individuele woorden, die net als bij het genereren van een zin worden geconditioneerd op de voorgaande woorden in die zin en de woorden die zijn gedetecteerd door de woorddetector. De log-waarschijnlijkheid van een zin is de som van de log-waarschijnlijkheden van de losse woorden. Vervolgens is de totale log-waarschijnlijkheid van alle zinnen weer de som van die van de losse zinnen. Het trainen van het model bestaat dus uit het maximaliseren van deze totale log-waarschijnlijkheid.

In plaats van één zin te genereren, houdt het model een stack bij van gedeeltelijke zinnen, terwijl het door middel van beam search bij elke stap alle zinnen uitbreidt met een verzameling van de meest waarschijnlijke woorden. Vervolgens worden de $k$ beste zinnen bewaard, en de rest wordt weggesnoeid. Als dit proces voltooid is en een verzameling met volledige kandidaatzinnen heeft voortgebracht, worden alle zinnen die minder dan $M$ van de gedetecteerde woorden bevatten, weggegooid. Vervolgens wordt de overgebleven lijst gesorteerd in aflopende volgorde, op basis van de log-waarschijnlijkheid van de zinnen.

In de laatste stap van het model van Fang et al. worden de zinnen opnieuw gesorteerd aan de hand van een aantal eigenschappen, waaronder de log-waarschijnlijkheid en de lengte van de zinnen. Een extra eigenschap die in deze stap wordt meegenomen, is de score van een Deep Multimodal Similarity Model (DMSM). Net als in de multimodale ruimte van het model van Karpathy & Fei-Fei (2015), projecteert dit model beeld en tekst naar een gezamenlijke vectorrepresentatie. Vervolgens wordt er een score berekend die de overeenkomst tussen de zin en de afbeelding weergeeft. Als de zinnen opnieuw zijn gesorteerd, wordt de beste zin gekozen als beschrijving van de inputafbeelding.

# 4. Prestaties modellen

In dit hoofdstuk bespreek ik de prestaties van de eerder genoemde AIC-modellen en zal ik proberen deze prestaties met elkaar te vergelijken, wat niet altijd even gemakkelijk is aangezien de gebruikte datasets en evaluatiemechanismen per model verschillen wat betreft inhoud en werking. In de eerste twee paragrafen geef ik een overzicht van veelgebruikte datasets en evaluatiemechanismen. Vervolgens beschrijf ik in paragraaf 4.3 de prestaties van de modellen aan de hand van de testresultaten zoals die in de papers zelf vermeld staan. Tot slot vul ik deze informatie aan met de inzichten van twee papers die meerdere modellen met elkaar vergelijken.

### 4.1 Datasets

Sinds de opkomst van AIC is ook de behoefte aan grootschalige datasets toegenomen. Voor het trainen en testen van AIC-systemen zijn namelijk grote hoeveelheden afbeeldingen nodig die voorzien zijn van één of meer door mensen gegenereerde beschrijvingen. In de afgelopen zes jaar is het aantal beschikbare fotobeschrijving datasets dan ook sterk toegenomen. Een aantal voorbeelden van veelgebruikte sets zijn Pascal1K, Flickr8K, Flickr30K en Microsoft COCO, die ik in deze paragraaf zal introduceren.

De PASCAL VOC-2008 dataset (Rashtchian et al., 2010, Hodosh et al., 2013), die ook wel Pascal1K wordt genoemd, is een set van 1000 afbeeldingen die afkomstig zijn uit de VOC-2008 trainingset van de Pattern Analysis, Statistical Modeling, and Computational Learning (PASCAL) organisatie. Om de afbeeldingen van beschrijvingen te voorzien, hebben de samenstellers Amazon Mechanical Turk ingezet. Mechanical Turk is een online platform waar gebruikers tegen een kleine betaling mee kunnen doen aan onderzoeken door simpele testjes te doen of vragen te beantwoorden. Voor de Pascal1K dataset werden per afbeelding vijf beschrijvingen verzameld, die na eventuele correcties werden opgenomen in de set.

In dezelfde papers presenteren de makers van de Pascal1K een tweede dataset: Flickr8K. Deze set bevat zo'n 8000 afbeeldingen afkomstig van Flickr, met elk vijf verschillende beschrijvingen, eveneens verzameld via Mechanical Turk. In tegenstelling tot Pascal1K, die ook foto's van statische situaties bevat, bestaat Flickr8K uitsluitend uit foto's van acties en gebeurtenissen. Volgens dezelfde richtlijnen hebben Young et al. (2014) Flickr8K uitgebreid tot de Flickr30K dataset, die ruim 30.000 geannoteerde afbeeldingen bevat.

Microsoft Common Objects in Context (Chen et al., 2015, Lin et al., 2014, MS COCO) is verreweg de meest uitgebreide van de hier genoemde datasets. Deze dataset telt 82.783 trainingsafbeeldingen, 40.504 validatieafbeeldingen en 40.775 testafbeeldingen. Bovendien zijn de afbeeldingen niet alleen voorzien van vijf beschrijvingen per stuk, maar ook van gedetailleerde labels en contouren voor alle belangrijke objecten.

### 4.2 Evaluatiemechanismen

Als een AIC-model eenmaal getraind is op een bepaalde dataset, moeten de prestaties ervan worden geëvalueerd. In het algemeen wordt de score van een model bepaald door de geproduceerde beschrijvingen te vergelijken met de menselijke beschrijvingen uit de testset. Hoe dichter de automatisch gegenereerde beschrijvingen in de buurt komen van de referentiebeschrijvingen, hoe hoger de score. Een voor de hand liggende manier om deze score te bepalen is om hiervoor mensen in te zetten. Hoewel menselijke beoordeling erg nauwkeurig en betrouwbaar is, is dit proces ook erg tijdrovend. Daarom geeft men soms de voorkeur aan auto-

matische evaluatiemechanismen, die vaak redelijk betrouwbare resultaten geven en bovendien veel sneller zijn. Van deze automatische evaluatiemechanismen geef ik in deze paragraaf een aantal voorbeelden.

BLEU (Papineni et al., 2002) is een mechanisme dat oorspronkelijk is geïntroduceerd om machinevertalingen te evalueren. Op basis van n-gram precision (meestal worden 1- t/m 4-grams gebruikt) tussen een machinevertaling en professionele vertalingen wordt aan de machinevertaling een 'translation closeness'-score toegekend. Door van alle vertaling binnen een corpus het gemiddelde van de score te nemen, wordt de score van het gehele corpus verkregen. Behalve bij machinevertalingen blijkt BLEU ook bruikbaar binnen AIC. Tegenwoordig wordt BLEU dan ook veelvuldig ingezet om de automatisch gegenereerde beschrijvingen te beoordelen, waarbij de referentiebeschrijvingen uit de testset de rol van de referentievertalingen vervullen.

Net als BLEU is ook METEOR (Banerjee & Lavie, 2005) ontwikkeld voor het beoordelen van machinevertalingen. METEOR maakt gebruik van unigram precision en unigram recall om een score te berekenen, waarbij niet alleen exact gelijke woorden worden gematcht, maar ook verbuigingen van hetzelfde woord en synoniemen.

Recentelijk presenteerden Vedantam et al. (2015) een nieuw evaluatiemechanisme: CIDEr. In tegenstelling tot de vorige twee mechanismen is CIDEr speciaal ontwikkeld voor het evalueren van beschrijvingen binnen AIC. Eerst worden alle woorden in de kandidaatbeschrijvingen en referentiebeschrijvingen gemapt naar hun stam of basisvorm. Vervolgens worden voor verschillende n-gramlengtes op basis van precision en recall de scores berekend. Net als bij BLEU blijkt het gebruik van 1- t/m 4-grams het beste te werken. Uiteindelijk combineert CIDEr de scores voor deze verschillende n-grams om tot een score voor een zin te komen.

### 4.3 Resultaten

Fang et al. (2015) hebben hun model getraind en getest op de MS COCO dataset. Omdat voor de testset geen beschrijvingen beschikbaar zijn, hebben de auteurs de validatieset opgedeeld in een validatieset en een testset. Het model werd getest met een aantal automatische mechanismen: PPLX, BLEU, en METEOR. PPLX (perplexiteit) staat grofweg voor het aantal keuzemogelijkheden en dus de onzekerheid van een model, waarbij een lagere perplexiteit dus een hogere score betekent. Daarnaast is het model getest door menselijke beoordelaars via Mechanical Turk. In de testresultaten worden de scores van zeven verschillende varianten van het model vermeld. Deze varianten verschillen o.a. in welke CNN als basis is gebruikt voor de objectdetector (AlexNet of VGGNet, zie hoofdstuk 2) en of de DMSM-score wel of niet is meegenomen in de hersorteerstap (zie paragraaf 3.3). De beste resultaten werden bereikt met de combinatie 'VGG+Score+DMSM+ft': hierbij wordt de VGG CNN als basis voor de woorddetector gebruikt, en worden de DMSM-scores meegenomen in de hersorteerstap. Deze variant van het model heeft een perplexiteit van 18.1, een BLEU-score van 25.7 (≤ 4-grams), en een METEOR-score van 23.6. Menselijke beoordelaars beoordelen de beschrijvingen die dit model produceert in 34.0% van de gevallen als 'beter dan of gelijk aan' door mensen geproduceerde beschrijvingen.

Karpathy & Fei-Fei (2015) gebruiken de Flickr8K, Flickr30K, en MS COCO datasets. Ze testen twee versies van hun model met BLEU, METEOR, en CIDEr. De eerste versie gebruikt de 16-lagige VGG-CNN van Simonyan & Zisserman (2014) en is getraind om zinnen te genereren bij volledige afbeeldingen. Dit model haalt op de MS COCO dataset een BLEU-score van 23.0 (≤ 4-grams), een METEOR-score van 19.5, en een CIDEr-score van

66.0. De tweede versie is getraind op de overeenkomst tussen regio's van afbeeldingen en stukjes tekst. Om dit model te testen hebben de auteurs via Mechanica Turk 200 afbeeldingen uit de MS COCO dataset laten voorzien van zo'n 9000 objectkaders die elk gelabeld zijn met een beschrijvend stukje tekst (gemiddeld 2.3 woorden per stuk). De vermelde BLEU-scores zijn 35.2, 23.0, 16.1, en 14.8 voor respectievelijk 1-, 2-, 3- en 4-grams.

Elliot & De Vries (2015) voeren hun experimenten uit op twee datasets: Pascal1K en VLT2K. VLT2K is een dataset van 2.424 afbeeldingen die specifiek acties uitbeelden, elk met drie beschrijvingen die via Mechanical Turk zijn verkregen. De gebruikte evaluatiemechanismen zijn METEOR en BLEU. Bovendien worden de testresultaten vergeleken met die van de MRNN van Karpathy & Fei-Fei (2015). Het meest opvallende resultaat is dat het model van Elliot & De Vries op de VLT2K dataset (14.8 BLEU, 16.0 METEOR) significant beter presteert dan op Pascal1K (9.0 BLEU, 7.4 METEOR). Ook gaat het model gelijk op met de MRNN als het om de VLT2K dataset gaat, maar blijft het op de Pascal1K set ver achter. Een mogelijke verklaring hiervoor is het feit dat de spatiële relaties die de VDR-parser detecteert, voornamelijk relevant zijn bij het beschrijven van foto's van dynamische taferelen.

Devlin et al. (2015) vergelijken in hun paper een aantal taalmodellen, waaronder het Maximum Entropy model van Fang et al. (2015), en een eigen implementatie van een Multimodaal Recurrent Neuraal Netwerk, vergelijkbaar met het model van Karpathy & Fei-Fei (2015). De gebruikte dataset is MS COCO en beide taalmodellen worden getraind op de output van de 16-lagige VGG CNN van Simonyan & Zisserman (2014). Na een eerste vergelijking blijkt dat de MRNN de ME LM (zonder DMSM) verslaat op zowel perplexiteit (13.2/18.1) als BLEU-score (25.7/23.6). De METEOR-score is voor beide modellen vrijwel gelijk (22.6/22.8).

Bij een tweede vergelijkingsronde testen de auteurs verschillende varianten en combinaties van modellen. Ook worden naast de automatische evaluatiemechanismen menselijke beoordelaars ingezet. Opmerkelijk is dat een combinatie van de MRNN en de ME LM, dit keer inclusief de DMSM-scores, de beste BLEU-score oplevert: 27.3. Helaas laat dit zich niet terugzien in een significante verbetering van de menselijke beoordeling: 34.2% van de zinnen die dit combinatiemodel produceert, worden beoordeeld als 'beter dan of gelijk aan' de door mensen geproduceerde alternatieven. Bij de best presterende versie van het ME LM alleen was dit nagenoeg gelijk: 34.0%.

Bernardi et al. (2016) geven in hun paper een meer oppervlakkige vergelijking van een groter aantal modellen. Het model van Karpathy & Fei-Fei komt in deze evaluatie uit de bus als één van de best presterende modellen van de afgelopen jaren. Echter, zoals de auteurs terecht opmerken, zijn de multimodale vergelijkingsmechanismen die dit soort modellen gebruiken, moeilijk te definiëren en vereisen ze een zeer grote trainingsset.

# 5. Discussie

Het gebied van AIC staat nog in de kinderschoenen. Dit betekent dat er nog een groot aantal nieuwe uitdagingen liggen en er veel onderzocht en verbeterd kan worden. Naast het verbeteren van de huidige prestaties kunnen ook de functies uitgebreid worden. In dit hoofdstuk stel ik een aantal richtingen voor waarin de huidige systemen uitgebreid of verbeterd kunnen worden.

Een punt waar verbetering mogelijk is, is de originaliteit van de gegenereerde zinnen. Karpathy & Fei-Fei vermelden bijvoorbeeld dat 60% van de door hun model gegenereerde

zinnen letterlijk voorkomt in de trainingsset, indien beam search wordt gebruikt met een beam size van 7. Ondanks dat het niet om description retrieval gaat, maar om description generation, heeft het model blijkbaar de neiging om bepaalde uitdrukkingen en zelfs hele zinnen letterlijk over te nemen uit de trainingsset. Een mogelijke oorzaak is te weinig variatie en specificiteit in de beschrijvingen van de trainingsset. Naast het feit dat de beschrijvingen van het model van Fang et al. gemiddeld als beste worden beoordeeld door mensen (zie paragraaf 4.3), scoort het volgens Devlin et al. (2015) ook beter wat betreft originaliteit: hier was slechts 30% van de zinnen letterlijk terug te vinden in de trainingsset. Dit wijst erop dat kwaliteit en originaliteit van de beschrijvingen elkaar niet per definitie uitsluiten. Toch scoren door mensen gegenereerde beschrijvingen nog altijd het beste: hiervan was slechts 4.8% een exacte kopie van een zin uit de trainingsset. Op dit punt is dus nog verbetering mogelijk in de toekomst.

Een andere moeilijkheid bij de huidige systemen is de afhankelijkheid van grote, door mensen gelabelde datasets. Het samenstellen van deze datasets is een erg arbeidsintensieve taak, wat tot gevolg heeft dat er maar enkele bruikbare exemplaren beschikbaar zijn, en veel onderzoeken dus afhankelijk zijn van hetzelfde, beperkte aanbod. Het zou de diversiteit aan middelen vergroten als we automatisch afbeeldingen met beschrijvingen uit het internet konden extraheren. Tegelijkertijd moet ik erkennen dat dit een erg lastige opgave is, aangezien het niet gebruikelijk is om afbeeldingen op het internet van een letterlijke beschrijving te voorzien zoals een AIC-systeem die nodig heeft, simpelweg omdat deze beschrijvingen precies vertellen wat voor ons mensen vanzelfsprekend is.

De modellen die in deze scriptie worden besproken, worden getraind op willekeurige foto's die willekeurige mensen en locaties afbeelden. Dit betekent dat de bijbehorende referentiebeschrijvingen alle personen en objecten met algemene termen zullen omschrijven. Dit heeft op zijn beurt weer tot gevolg dat de objectdetectors getraind worden om alle objecten in deze algemene categorieën in te delen. Een foto van Martin Luther King Jr. of een foto van de Dam in Amsterdam zullen bijvoorbeeld niet als zodanig herkend worden: een objectdetector zal hoogstens een 'man' en een 'plein' herkennen. Een nuttige uitbreiding zou dus zijn om de AIC-technologie te combineren met andere technologieën zoals gezichtsherkenning, zodat niet alleen categorieën, maar ook specifieke instanties van objecten, personen en locaties kunnen worden herkend. Dit brengt uiteraard wel weer de behoefte aan grotere en uitgebreidere datasets met zich mee.

# 6. Conclusie

In deze scriptie heb ik het onderzoeksgebied van automatic image captioning aan de lezer geïntroduceerd en een overzicht gegeven van veel toegepaste strategieën en specifieke modellen. Waar de meeste modellen een vergelijkbaar globaal stappenplan volgen, blijkt de implementatie onderling te verschillen.

Sommige modellen detecteren woorden in de afbeelding en bouwen aan de hand van deze woorden een zin op. Andere modellen projecteren tekst en beeld in een gezamenlijke multimodale ruimte en conditioneren het generatieproces van de beschrijving op de representatie van de inputafbeelding. Ten slotte zijn er nog modellen die een extra component bevatten die spatiële relaties tussen objecten in kaart brengt, en deze vervolgens gebruikt om kwalitatieve beschrijvingen te produceren.

Ook verschillen de prestaties per model. Van de hier beschreven modellen lijkt het model van Fang et al. (2015), dat gebruik maakt van een maximum entropy taalmodel, de beste

prestaties te leveren volgens menselijke beoordeling, mits de DMSM-score wordt meegenomen in de hersorteerstap.

Ten slotte heb ik een aantal suggesties gedaan met betrekking tot eventuele verbeteringen en uitbreiding van functies. Zo is er nog verbetering mogelijk in de originaliteit van de gegenereerde beschrijvingen, en zouden AIC-systemen kunnen worden gecombineerd met andere technologieën zoals gezichtsherkenning, zodat ook specifieke personen of locaties herkend kunnen worden.

# Geraadpleegde literatuur


Banerjee, S., & Lavie, A. (2005). METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *Annual Meeting of the Association for Computational Linguistics Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization*.

Berger, A.L., Della Pietra, S.A., & Della Pietra, V.J. (1996). A Maximum Entropy Approach to Natural Language Processing. *Computational Linguistics, 22(1)*, 39-71.

Bernardi, R., Cakici, R., Elliott, D., Erdem, A., Erdem, E., Ikizler-Cinbis, N., Keller, F., Muscat, A., & Plank, B. (2016). Automatic Description Generation from Images: A Survey of Models, Datasets, and Evaluation Measures. In *Journal of Artificial Intelligence Research, 55*, 409-442.

Chen, X., Fang, H., Lin, T.-Y., Vedantam, R., Gupta, S., Dollár, P., Zitnick, C.L. (2015). Microsoft COCO Captions: Data Collection and Evaluation Server. *arXiv preprint arXiv:1504.00325v2*.

Devlin, J., Cheng, H., Fang, H., Gupta, S., Deng, L., He, X., Zweig, G., & Mitchell, M. (2015). Language Models for Image Captioning: The Quirks and What Works. In *Annual Meeting of the Association for Computational Linguistics*.

Dietterich, T.G., Lathrop, R.H., & Lozano-Pérez, T. (1997). Solving the multiple instance problem with axis-parallel rectangles. In *Artificial Intelligence, 89(1-2)*, 31-71.

Duygulu, P., Barnard, K., Freitas, J.F.G. de, & Forsyth, D.A. (2002). Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *Seventh European Conference on Computer Vision, 4*, 97-112.

Elliott, D., & Vries, A.P. de (2015). Describing images using inferred visual dependency representations. In *Annual Meeting of the Association for Computational Linguistics*.

Fang, H., Gupta, S., Iandola, F., Srivastava, R., Deng, L., Dollár, P., Gao, J., He, X., Mitchell, M., Platt, J., Zitnick, C. L., & Zweig, G. (2015). From captions to visual concepts and back. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Girshick, R., Donahue, J., Darrell, T., & Malik., J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. *arXiv preprint arXiv:1311.2524v5*.

Hodosh, M., Young, P., & Hockenmaier, J. (2013). Framing Image Description as a Ranking Task: Data, Models and Evaluation Metrics. *Journal of Articial Intelligence Research, 47*, 853-899.

Karpathy, A., & Fei-Fei, L. (2015). Deep visual-semantic alignments for generating image descriptions. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Krizhevsky, A., Sutskever, I., & Hinton, G.E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In *NIPS*.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C.L. (2014). Microsoft COCO:

Common Objects in Context. In *European Conference on Computer Vision*.

Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., & Khudanpur, S. (2010). Recurrent neural network based language model. In *INTERSPEECH 2010*, 1045-1048.

Mori, Y., Takahashi, H., & Oka, R. (1999). Image-to-word transformation based on dividing and vector quantizing images with words. In *MISRM'99 First International Workshop on Multimedia Intelligent Storage and Retrieval Management*.

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). BLEU: a Method for Automatic Evaluation of Machine Translation. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, 311-318.

Rashtchian, C., Young, P., Hodosh, M., & Hockenmaier, J. (2010). Collecting Image Annotations Using Amazon's Mechanical Turk. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk.*

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Vedantam, R., Zitnick, C.L., & Parikh, D. (2015). Cider: Consensus-based image description evaluation. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Young, P., Lai, A., Hodosh, M., & Hockenmaier, J. (2014). From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics, 2*, 67-78.