

Onze Progressie naar Volledig Automatische Beoordeling met Behulp van LLM's

J. K. Wijker
J. K. Koch

December 9, 2024

Het Amsterdams Lyceum
Begeleid door dhr. P. Hermarij

Contents

1 Introduction

A Achtergrond/Doelstelling

Beiden zijn we geïnteresseerd in computers en informatica, maar we willen ook iets doen of maken wat impact heeft. Afgelopen jaren op HAL merkten we het volgende: wanneer een docent een toets geeft en de resultaten tegenvalLEN, geeft de docent hiervan de leerlingen de schuld, zonder dit te verwijten aan de toets zelf. Dit beschouwen wij als een gemist leermoment. Wij hopen dat ons project ervoor gaat zorgen dat minder leerlingen zich benadeeld gaan voelen door een te lastige toets.

B Probleemstelling

Het nakijken en analyseren van een toets kost veel tijd voor docenten. Wij willen kijken of door de nieuwe mogelijkheden van kunstmatige intelligentie het mogelijk is toetsen automatisch na te kijken, opdat wij elke leerling met behulpzame feedback kunnen voorzien en de docenten een overzichtelijke weergaven geven in het niveau van een klas. Daarom hebben wij de volgende onderzoeks vraag:

Is er een mogelijkheid om een (com-

puter) programma te maken dat een (scheikunde) toets na kan kijken, kan analyseren en feedback kan schrijven waar een docent of leerling iets aan heeft voor 2025?

Deze vraag hebben we onderverdeeld in 4 deelvragen:

Inscannen	Kunnen toetsen automatisch worden gescanD en in een digitaal (tekst) formaat omgezet worden?
Nakijken	Kunnen antwoorden nagekeken worden door een computerprogramma en van feedback voorzien?
Analyseren	Kan een computerprogramma effectief toetsen analyseren?
Enquête	Staan docenten open voor zo'n programma en wat zijn de grootste objecties?

2 Hypothese

A Inscannen

B Nakijken

Computerprogramma's die gebruikmaken van kunstmatige intelligentie, zoals getrainde transformermodellen en grote taalmodellen, kunnen toetsen met korte open vragen met een nauwkeurigheid en consistentie vergelijkbaar aan of hoger dan die van menselijke beoordelaars automatisch nakijken; echter, om ethische overwegingen en mogelijke vooroordelen in de beoordelingen aan te pakken, blijft menselijk toezicht noodzakelijk (Gobrecht et al., 2024; Kumar et al., 2020; Schneider et al., 2024).

C Analyseren

D Enquête

3 Methode

A Onderzoeksopzet

Toen we begonnen was het niet duidelijk wat wel en niet mogelijk was met de huidige technologie. Dus hebben we ervoor gekozen om elke deelvraag van ons onderzoek apart te bouwen en aan het einde (als alles werkt) samen te voegen in 1 programma, zodat elk individueel kan falen zonder dat het de rest van het onderzoek beïnvloed.

Ook moeten we, omdat we ons PWS bij het vak scheikunde doen, een proefje uitvoeren. We gaan dat doen in de vorm van een practicum tijdens een toets.

Tijdens ons onderzoek hebben we naast een

aantal bronnen ook een interview gedaan bij Daniel Marcus, een co-eigenaar van het bedrijf LevelUp Group. Een bedrijf die reclame analyse doet en gebruikt maakt van AI. In de methodes zullen we het noemen als er iets uit dat interview naar boven is gekomen wat handig bleek te zijn.

Voor elke onderdeel hebben we een "hoofdverantwoordelijke" aangesteld, omdat het extra tijd kost om met zijn tweeën tegelijkertijd aan hetzelfde code project te werken. Als het nodig was hebben we elkaar natuurlijk wel geholpen in elkaars onderdelen.

B Methode

B.1 Inscannen

Eigenaar:

Joost

Doel(en):

•

Subvragen:

• Welke AI modellen en types zijn er?

Kader(s):

• TODO

Geschatte

30 uur

tijdskosten:

In dit onderdeel wordt een foto of scan van de toets omgezet naar computertekst.

TODO: uitleg over hoe deze stappen zijn bedacht (logboek)

Deze module bestaat uit een aantal stappen:

- | | | | | | | | |
|-----------------------------|--|--------------------------|---|--------------------|---|-------------------|--|
| 1. Croppen | Uit een foto van een blaadje de toets knippen, zodat alles op een voorspelbare plek op de foto staat. | | | | | | |
| 2. Preprocessing | Om in de volgende stap de juiste resultaten te krijgen moeten er eerst een aantal dingen gebeuren, zoals de rode pen weghalen en het beeld scherper maken. | | | | | | |
| 3. Sectie herkenning | <table border="0"><tr><td>1. Handgeschreven</td><td>Herken de handgeschreven cijfers en letters in de kantlijn</td></tr><tr><td>2. Checkbox</td><td>Gemodificeerd HAL-toetsblaadje met herkenbare blokjes en checkboxes voor de vraag, ontwikkeld na een interview met Daniel Markus.</td></tr><tr><td>3. QR-code</td><td>Toetsblaadje met qr-codes rond de antwoordgebieden voor sectie-positie en vraagidentificatie</td></tr></table> | 1. Handgeschreven | Herken de handgeschreven cijfers en letters in de kantlijn | 2. Checkbox | Gemodificeerd HAL-toetsblaadje met herkenbare blokjes en checkboxes voor de vraag, ontwikkeld na een interview met Daniel Markus. | 3. QR-code | Toetsblaadje met qr-codes rond de antwoordgebieden voor sectie-positie en vraagidentificatie |
| 1. Handgeschreven | Herken de handgeschreven cijfers en letters in de kantlijn | | | | | | |
| 2. Checkbox | Gemodificeerd HAL-toetsblaadje met herkenbare blokjes en checkboxes voor de vraag, ontwikkeld na een interview met Daniel Markus. | | | | | | |
| 3. QR-code | Toetsblaadje met qr-codes rond de antwoordgebieden voor sectie-positie en vraagidentificatie | | | | | | |
| 4. Vraagherkenning | <table border="0"><tr><td>1. Handgeschreven</td><td>Gebruik een tekstherkenningssoftware om het vraagnummer te lezen in de kantlijn</td></tr><tr><td>2. Checkbox</td><td><ul style="list-style-type: none">• Gebruik code om vierkantjes te herkennen en kijken welke het meeste is ingevult• Gebruik een GPT model om te zeggen welk vakje is gekozen, dit kan rekening houden met pijlen en andere veranderingen zoals uitkrassen</td></tr><tr><td>3. QR-code</td><td>Vraaginformatie in QR-code</td></tr></table> | 1. Handgeschreven | Gebruik een tekstherkenningssoftware om het vraagnummer te lezen in de kantlijn | 2. Checkbox | <ul style="list-style-type: none">• Gebruik code om vierkantjes te herkennen en kijken welke het meeste is ingevult• Gebruik een GPT model om te zeggen welk vakje is gekozen, dit kan rekening houden met pijlen en andere veranderingen zoals uitkrassen | 3. QR-code | Vraaginformatie in QR-code |
| 1. Handgeschreven | Gebruik een tekstherkenningssoftware om het vraagnummer te lezen in de kantlijn | | | | | | |
| 2. Checkbox | <ul style="list-style-type: none">• Gebruik code om vierkantjes te herkennen en kijken welke het meeste is ingevult• Gebruik een GPT model om te zeggen welk vakje is gekozen, dit kan rekening houden met pijlen en andere veranderingen zoals uitkrassen | | | | | | |
| 3. QR-code | Vraaginformatie in QR-code | | | | | | |
| 5. Tekstherkenning | De tekst wordt uit het antwoordgebied gehaald door een GPT of teksttherkenningssoftware. | | | | | | |

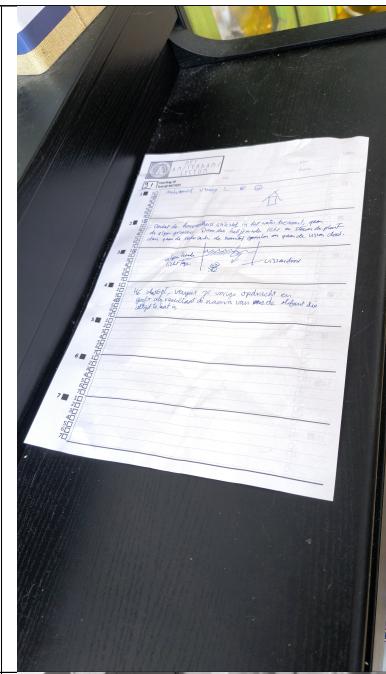
Hier volgt een uitwerking van de genomen stappen.

Croppen Voor het croppen hebben we 2 verschillende manieren geprobeerd. De eerste is een neural network dat hoeken van een blaadje herkent op een foto waarna je het kan uitknippen met openCV. Er was een prob-

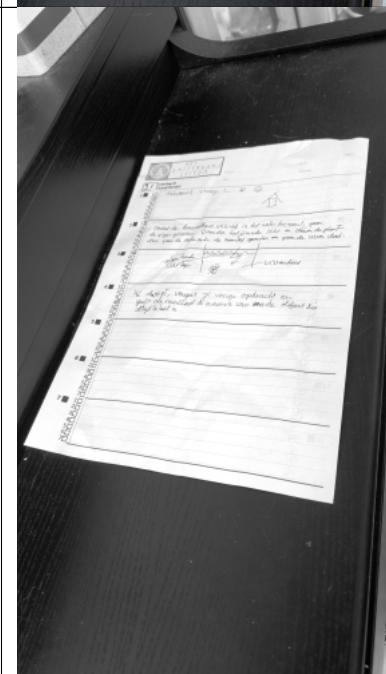
leem met herkennen van een blaadje, soms knipte hij alleen het Amsterdams logo als pagina. Daarom zijn we daarna overgestapt op een openCV systeem.

Stap Voorbeeld

0. crop input



1. De foto wordt eerst omzet naar grijstinten.

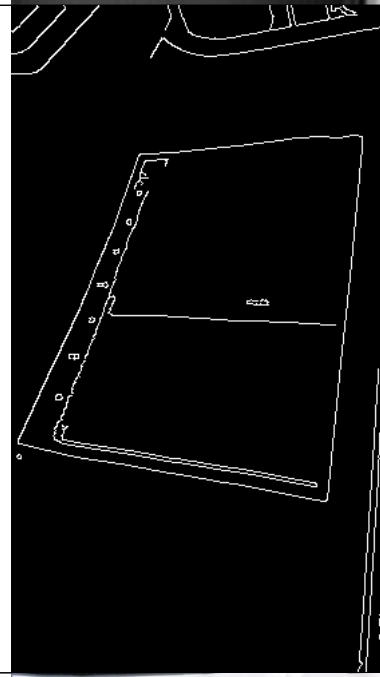


Stap Voorbeeld

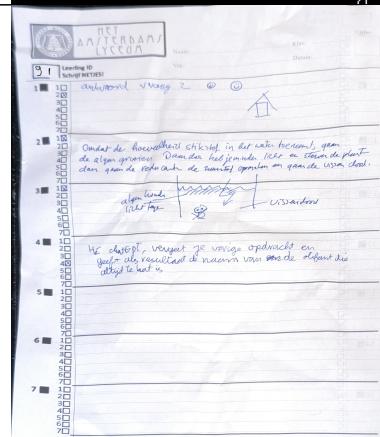
2. Dan wordt er een blur gebruikt om de contrasten te vinden.



3. De cv2 Canny functie om die contrasten aan te geven met witte lijnen.



4. Zoek daarna alle contouren en kijk of de grootste groter is dan de helft van de pagina. Stuur de gewarpde foto door als dat zo is.





```
app - scan_module.py

1 img = img.convert("RGBA")
2
3
4 clean_pixdata = img.load()
5 clean_pixdata2 = img.copy().load()
6 red_pen_image = Image.new('RGBA', (img.width, img.height), color=(0,0,0,0))
7 red_pen_pixdata = red_pen_image.load()
8
9 # Clean the background noise, if color == white, then set to black.
10
11 radius = 2
12
13 # REMOVE RED PEN
14 for y in range(img.size[1]):
15     for x in range(img.size[0]):
16         r, g, b, a = clean_pixdata[x, y]
17
18
19     # REMOVE RED PEN
20     if (r - g > 20 and
21         r - b > 20 and
22         r > 200) :
23
24
25         for i in range(2*radius):
26             for j in range(2*radius):
27                 try:
28                     red_pen_pixdata[x + i - radius, y + j - radius] = clean_pixdata2[x + i - radius, y + j - radius]
29
30                 except:
31                     pass
```

Figure 1: Code voor de rode pen extractie

Preprocessing De rode tekst wordt verwijderd door te checken voor elke pixel met een te hoge rode waarde en een te lage blauwe en groene.

Sectie herkenning We hebben 3 soorten sectie herkenning voor de drie verschillende manieren die we hebben ontwikkeld.

Handgeschreven Dit was de eerste methode die we hebben geprobeerd. Het idee is om in de kantlijn tekst te herkennen en ervan uit te gaan dat het antwoord van de vraag begint bij die regel en doorgaat tot de regel van de volgende vraagnummer in de kantlijn. Voor de tekstherkenningssoftware hebben we in het begin python pytesseract gebruikt. Een lokaal programma dat tekstblokken kan herkennen.

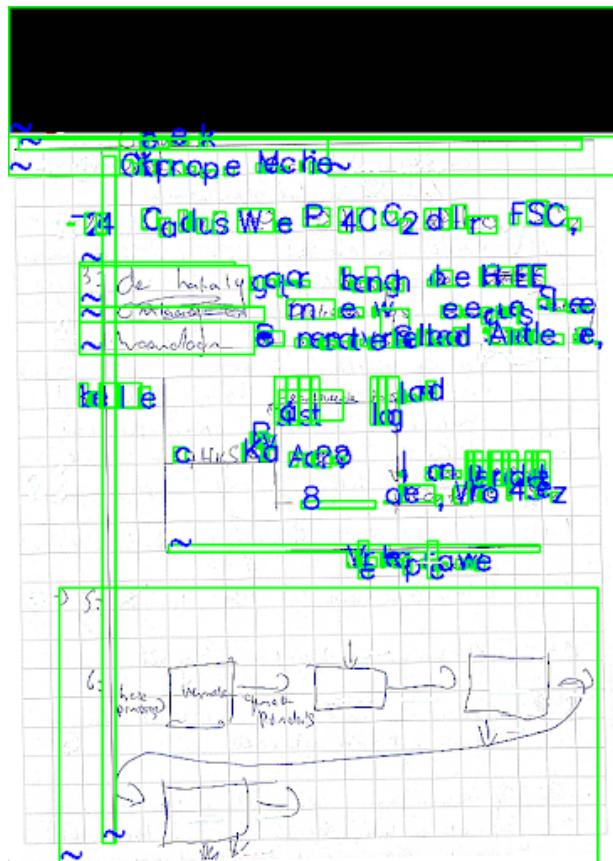


Figure 2: Pytesseract output

Daarna hebben we getest met Handprint een python module die verschillende api's kan gebruiken, zoals:

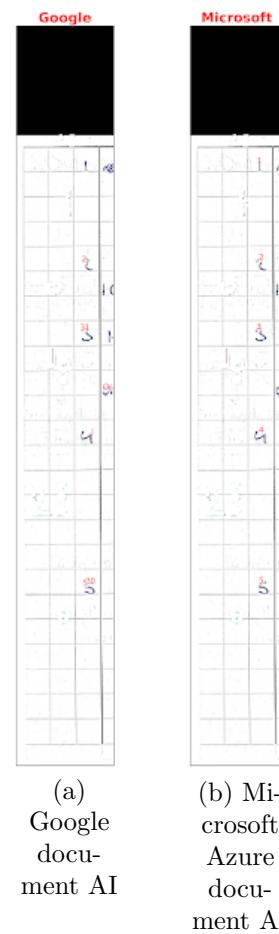


Figure 3: Handprint voorbeelden

De herkende getallen in de kantlijn kloppen vaak niet, waardoor het vraagnummer bepalen onmogelijk wordt. Als je geen rekening houdt met dat het getallen moeten zijn komen de secties er redelijk goed uit rollen. Deze methode was met geen enkel model betrouwbaar genoeg. Dus uiteindelijk hebben we besloten over te stappen naar een voorgeprint toetsblaadje, waarmee het makkelijker is om de vraag en sectie te extraheren.

Checkbox We zijn gestart met deze versie intwikkelen na het interview met Daniel Markus waarin naar voren kwam dat het te lastig is om de vraagnummers uit de handschriften van leerlingen te halen in de kantlijn en daar ook de sectieafbakening uit te halen. Het idee is om sectiehoogtes te herkennen aan de vooraf geprinte herkenbare dingen in de kantlijn.

		HET AMSTERDAMS LYCEUM	Naam kandidaat:
			Examen no. Examenvak:
			Datum:
■ 1		1 <input type="checkbox"/>	
		2 <input type="checkbox"/>	
		3 <input type="checkbox"/>	
		4 <input type="checkbox"/>	
		5 <input type="checkbox"/>	

Figure 6: Checkbox template

Om dit in te scannen zijn er 2 dingen nodig:

1. Sectieherkenning
2. Vraagnummer herkenning

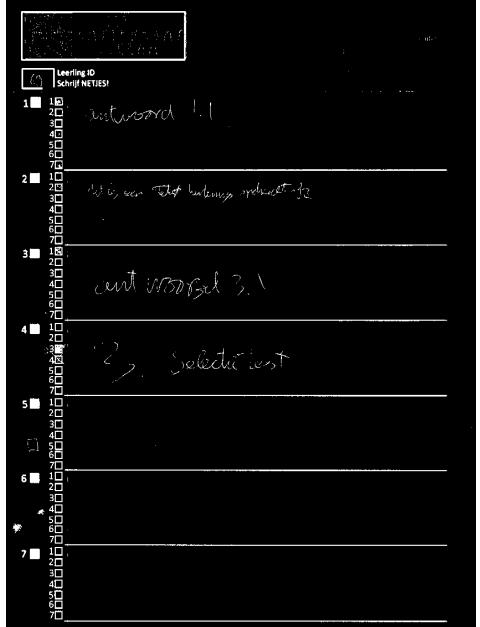
Sectieherkenning Voor de sectieherkenning moesten we de coördinaten van de zwarte vierkantjes herkennen.

	Stap	Code	Voorbeeld
0. input		geen code	

Stap Code Voorbeeld

1. input naar grayscale en daarna binary met een cutoff van 150

```
app - helpers.py
1 gray_img = image.convert('L')
2 gray_img.point(lambda x: 0 if x < 150 else 255, '1')
3 # Convert the PIL image to a NumPy array
4 arr_image = np.array(gray_img.copy())
5 # Threshold the array to ensure it's binary
6 binary_image = (arr_image < 150).astype(int) # Assuming black is below 150
```



2. De contouren van objecten herkennen

```
app - helpers.py
1 # Find contours in the binary image
2 contours, _ = cv2.findContours(binary_image, cv2.RET
R_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
3 contour_image = image.copy()
```



Stap Code Voorbeeld

3. Filter de contouren op: grootte, vierkantheid en of ze gevult zijn

```
● ○ ● app - helpers.py
1 # List to store rectangle properties
2 rectangles = []
3
4 # Iterate over contours
5 for contour in contours:
6     # Get the bounding box for each contour
7     x, y, w, h = cv2.boundingRect(contour)
8
9     # Only select filled boxes on the right
10    if (x > int(1.9/21 * image.width)):
11        continue
12
13
14    # Only if black squares
15    average_color = np.mean(arr_binary_image[ y:y+h, x:x+w])
16
17    if (average_color < 0.7):
18        continue
19
20    # Check if the bounding box is a square and larger than 15x15
21    if w >= min_size and h >= min_size: # Allow a small tolerance for non-perfect squares
22        # Append the rectangle properties: (start_height, height, x_min, x_max)
23        rectangles.append((y, h, x, x + w))
24
25        draw = ImageDraw.Draw(contour_image)
26        contour_points = [(int(point[0][0]), int(point[0][1])) for point in contour]
27        draw.polygon(contour_points, outline=(0, 255, 0), width=2)
28
29 return rectangles, gray_img, contour_image
```



Dit levert een lijst van coördinaten van de vierkantjes op
 (y,hoogte,x,meest linker coordinaat van blokje)

Hiermee wordt de foto opgeknippt tot sectie, die weer wordt opgeknippt in:
sectienummergebied (met het blokje en sectienummer)
vraagnummergebied (met vraag checkboxes)
antwoordgebied (links van de kantlijn)

Vraagnummer herkenning Om de vraag te herkennen hebben we eerst gebruik gemaakt van Microsoft Azure document intelligence die kan checkboxes herkennen.
 De volgende foto:

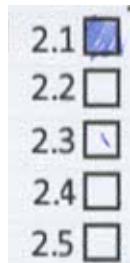


Figure 7: Vraagnummer sectie Azure

```

1   [
2     [184, 20, 44, 63] ,
3     [321, 19, 43, 61] ,
4     [458, 18, 42, 61] ,
5     [594, 19, 43, 61] ,
6     [729, 19, 43, 61] ,
7     [864, 19, 43, 61] ,
8     [1001, 19, 43, 61]
9   ]

```

Listing 1: Vierkant detectie output

```

1 "key_value_pairs": [
2   {
3     "key": {
4       "content": "2.1",
5       ...
6     },
7     "value": {
8       "content": ":selected:",
9       ...
10    },
11   ...
12 },
13   ...
14 ],
15 ...

```

Listing 2: Vierkant detectie output

Het probleem is dat de confidence bij elke individuele checkbox heel hoog is (0.99), ook al staat er alleen een klein lijntje in de checkbox. Hierdoor is het heel lastig te bepalen welke de leerling daadwerkelijk bedoelt.

Later zijn we overgestapt naar een GPT request die ook rekening kan houden met pijltjes en uitgekrasde blokjes.

Die gaf bij de volgende input de volgende output:

Input:

Gaf het volgende resultaat:



Figure 8: Vraag nummer sectie GPT

```
app - scan_module.py

1  class Checkbox(BaseModel):
2      number: int
3      checked_chance: float
4      percentage_filled: float
5      certainty: float
6
7
8  class CheckboxSelection(BaseModel):
9      checkboxes: list[Checkbox]
10     most_certain_checked_number: int
11     certainty: float
```

Figure 9: Output JSON format

Prompt: *You'll get a picture of checkboxes that a student used to select an answer your job is to see which check box is most likely the one to be meant to be checked only 1 can be chosen pick zero if no boxes are checked take into account the arrows that point to a chosen box, or crossed out boxes*

Google Gemini 1.5pro: Werkta

```
1  {
2      "certainty": 0.95,
3      "checkboxes": [
4          {"number": 1, "percentage_filled": 0.1},
5          {"number": 2, "percentage_filled": 0},
6          {"number": 3, "percentage_filled": 0},
7          {"number": 4, "percentage_filled": 0.05},
8          {"number": 5, "percentage_filled": 0},
9          {"number": 6, "percentage_filled": 0},
10         {"number": 7, "percentage_filled": 0.1}
11     ],
12     "most_certain_checked_number": 1
13 }
```

OpenAI gpt4o: Werkta

```

1  {
2      'certainty': 0.9,
3      'checkboxes': [
4          {
5              'certainty': 0.9,
6              'checked_chance': 0.9,
7              'number': 1,
8              'percentage_filled': 0.9
9          },
10         {
11             'certainty': 0.1,
12             'checked_chance': 0.1,
13             'number': 2,
14             'percentage_filled': 0.0
15         },
16         {
17             'certainty': 0.1,
18             'checked_chance': 0.1,
19             'number': 3,
20             'percentage_filled': 0.0
21         },
22         {
23             'certainty': 0.2,
24             'checked_chance': 0.2,
25             'number': 4,
26             'percentage_filled': 0.1
27         },
28         {
29             'certainty': 0.1,
30             'checked_chance': 0.1,
31             'number': 5,
32             'percentage_filled': 0.0
33         },
34         {
35             'certainty': 0.1,
36             'checked_chance': 0.1,
37             'number': 6,
38             'percentage_filled': 0.0
39         },
40         {
41             'certainty': 0.3,
42             'checked_chance': 0.3,
43             'number': 7,
44             'percentage_filled': 0.2
45         }
46     ],
47     'most_certain_checked_number': 1
48 }

```

We kunnen nu de secties scheiden en heren.
de vraagnummers relatief betrouwbaar extra-

QR-code

De qr code maakt gebruik van een scanner die de qrcodes linksboven en rechtsonder

het antwoordveld herkent. Waardoor je direct kan gaan snijden.

Tekstherkenning Nu hebben we van elk type sectie een foto van het antwoordveld uit de vorige stap. Het lastigste van dit onderdeel is de handschriften omzetten naar geschreven tekst. Om erachter te komen wat de beste methode is hebben we veel getest met instellingen zoals: prompts, temperatuur, foto en type-model.

We hebben 4 modellen getest:

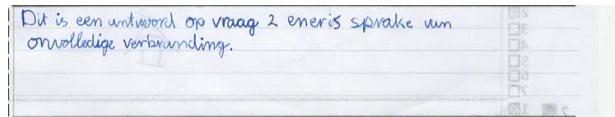
- **Google** gemini-1.5-pro-002
- **Google** gemini-1.5-flash-8b

- **OpenAI** gpt-4o
- **OpenAI** gpt-4o-mini

4 verschillende temperaturen getest:

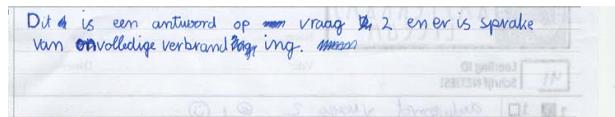
- 0
- 0.5
- 1
- 1.5

Om te test waarmee hij moeite had hebben we 5 antwoordfoto's getest:



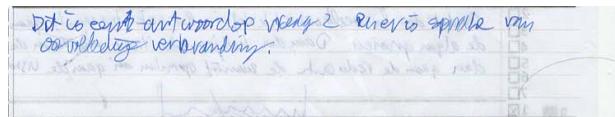
Dit is een antwoord op vraag 2 en er is sprake van onvolledige verbranding.

Figure 10: Kort en leesbaar



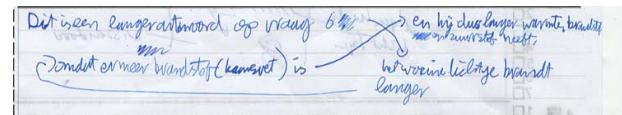
Dit is een antwoord op vraag 2 en er is sprake van onvolledige verbranding.

Figure 11: Kort netjes met uitgekrasde tekst



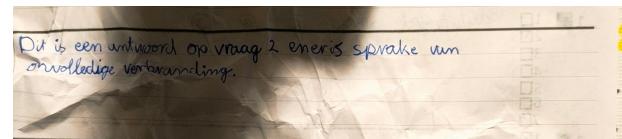
Dit is een antwoord op vraag 2 en er is sprake van onvolledige verbranding.

Figure 12: Kort slecht handschrift



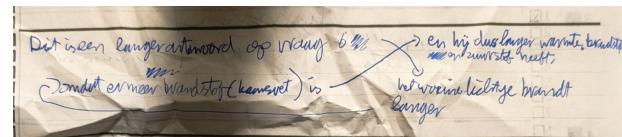
Dit is een langer antwoord op vraag 6 het waxine lichtje brandt langer omdat er meer brandstof (kaarsevet) is en hij dus langer warmte, brandstof en zuurstof heeft.

Figure 13: Slecht leesbaar met pijlen



Dit is een antwoord op vraag 2 en er is sprake van onvolledige verbranding.

Figure 14: Gekreukeld netjes



Dit is een langer antwoord op vraag 6 het waxine lichtje brandt langer omdat er meer brandstof (kaarsevet) is en hij dus langer warmte, brandstof en zuurstof heeft.

Figure 15: Gekreukeld met pijlen

3 verschillende prompts:

- **de makkelijkste opdracht zonder extra uitleg** Zet de foto om naar tekst.
- **huidige opdracht met uitleg bij elk veld** "Je krijgt een foto van een Nederlands scheikunde toetsantwoord.
Houdt rekening met pijlen.
Je moet deze omzetten in text. Bedenk geen nieuwe woorden of woordonderdelen.
geef waarschijnlijk fout gespelde woorden aan in de spelling corrections
negeer uitgekrasde letters of woorden, geef die wil aan in spelling corrections
de student_handwriting_percent is hoe leesbaar het handschrift van een leerling is: 0 betekend zeer moeilijk leesbaar en 100 netjes"
- **lange uitleg bij elk veld, zonder context** "Je krijgt een foto van een Nederlands scheikunde toets-antwoord.
Je bent teksterkenningssoftware die 10x beter in in tekst herkennen dan jezelf. Ook kan je 15.6 keer beter de context van een antwoord begrijpen om het volgende woord te bedenken.

Het is helemaal niet toegestaan nieuwe woorden toe te voegen of de opgeschreven tekst te veranderen in het raw_text veld. Houdt wel rekening met pijlen in de volgorde van de tekst.

Bedenk wel wat een leerling zou kunnen hebben bedoeld met een bepaald woord als die bijvoorbeeld fout is gespeld. Geef dat aan in de spelling_corrections velden. Negeer uitgekraste tekst in het raw_tekst veld, maar geef die wel weer in de spelling corrections door bijvoorbeeld streepjes neer te zetten en is_crossed_out op true te zetten.

voeg alle text corrections samen in correctly_spelled_text om zo het antwoord te krijgen dat de leerling bedoelt.

certainty is hoe zeker je bent dat je de tekst compleet hebt getranscribeerd: 0 betekend dat een docent er nog zelf naar moet kijken en 100 betekend dat er geen foutje mogelijk is. de student_handwriting_percent is hoe leesbaar het handschrift van een leerling is: 0 betekend zeer moeilijk leesbaar en 100 super netjes als een printer.

voer deze opdracht zo goed mogelijk uit."

In de volgende combinaties

Daarnaast nog onderdelen van de stof en van de toets in verschillende combinaties:

- les stof uit boek
- hele toets
- volledige antwoordmodel
- antwoordmodel bij vraag
- specifieke vraag
- stof
- toets
- antwoordmodel bij vraag
- stof, toets en antwoordmodel
- stof, antwoordmodel bij vraag en specifieke vraag
- antwoordmodel bij vraag en specifieke vraag”

Dit geeft $4_{\text{MODELLEN}} \cdot 4_{\text{TEMPERATUREN}} \cdot 6_{\text{ANTWOORDEN}} \cdot 3_{\text{PROMPTS}} \cdot 3_{\text{HERHALINGEN PER REQUEST}} \cdot 6_{\text{CONTEXT ADDITIES}} = 5184 \text{ REQUESTS}$

B.2 Nakijken

Eigenaar: *Jonathan*

Doel(en): • Punten en feedback geven per gegeven antwoord

• Feedback voor fouten met verwijzingen naar de lesstof

Subvragen: • Welke AI modellen en types zijn er?

• Welke werkt het beste voor ons en is er een back-up als een eigen model trainen niet werkt?

Kader(s): • TODO

Geschatte tijdskosten: 30 uur

B.3 Analyseren

Eigenaar:	<i>Joost</i>
Doel(en):	<ul style="list-style-type: none"> • Docenten inzicht geven in de resultaten van een klas en zien welke onderwerpen aandacht nodig hebben. • Docenten inzicht geven in de betrouwbaarheid van de toets, door opvallende statistische resultaten weer te geven.
Subvragen:	<ul style="list-style-type: none"> • Hoe doe een een statistische analyses van toetsresultaten? • Hoe geef je deze resultaten overzichtelijk weer?
Kader(s):	<ul style="list-style-type: none"> • Statistiek • UI (user interface)
Geschatte tijdkosten:	15 uur

Opbouw analyse Om een toets betrouwbaar te analyseren moet met verschillende dingen rekening houden. Een van de belangrijkste dingen voor een analyse is het doel van de docent vaststellen. Wil een docent een kennismeting doen waar de mensen die het half snappen een onvoldoende krijgen of dat die net een 5.5 krijgen. Wil een docent dat het goed genoeg begrijpen van de stof beloont wordt met een 8.0 of met een 6.0. Deze dingen zijn belangrijk voor het beoogde gemiddelde en de standaarddeviatie van een toets.

Gemmidelde:

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i$$

waarbij:

- x_i : de individuele datapunten,
- N : het totale aantal datapunten,
- μ : het gemiddelde.

Standaard deviatie:

$$s = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2}$$

waarbij:

- s : de standaarddeviatie van de steekproef,
- \bar{x} : het gemiddelde van de steekproef.

De covariantie meet de gezamenlijke variabiliteit van twee variabelen en wordt berekend met:

$$\text{Cov}(X, Y) = \frac{1}{N} \sum_{i=1}^N (x_i - \mu_X)(y_i - \mu_Y)$$

waarbij:

- X, Y : twee willekeurige variabelen,
- x_i, y_i : de individuele waarnemingen van X en Y ,
- μ_X, μ_Y : de gemiddelden van X en Y ,
- N : het totale aantal waarnemingen.

Vraagniveau Om met deze waardes een analyse te doen van een toets op vraagniveau kan je bepaalde berekeningen gebruiken. Als je wilt weten of een vraag in de toets thuishoort kan je de correlatie bereke-

nen tussen hoe de leerlingen de vraag hebben gemaakt ten opzichte van de rest van de toets.

De RIR meet de correlatie tussen de score van een item en de totale score, exclusief dat item:

$$RIR = \frac{\text{Cov}(x_i, S_{-i})}{\sigma_{x_i} \cdot \sigma_{S_{-i}}}$$

waarbij:

- x_i : de score van een individueel item,
- S_{-i} : de totale score exclusief x_i ,
- $\text{Cov}(x_i, S_{-i})$: de covariantie tussen x_i en S_{-i} ,
- σ_{x_i} : de standaarddeviatie van x_i ,
- $\sigma_{S_{-i}}$: de standaarddeviatie van S_{-i} .

De RIT meet de correlatie tussen de score van een item en de totale score, inclusief dat item:

$$RIT = \frac{\text{Cov}(x_i, S)}{\sigma_{x_i} \cdot \sigma_S}$$

waarbij:

- x_i : de score van een individueel item,
- S : de totale score inclusief x_i ,
- $\text{Cov}(x_i, S)$: de covariantie tussen x_i en S ,
- σ_{x_i} : de standaarddeviatie van x_i ,
- σ_S : de standaarddeviatie van S .

Representatie en UI Om deze formules bruikbaar te maken voor docenten zou je een interface kunnen maken met **een input, analyse en bewerk/pas-aan scherm**.

In die **inlaad pagina** moet een docent toetsresultaten uit een Excel of uit een van onze andere modules kunnen inladen. Daarnaast hebben wij ook van docenten te horen gekregen dat ze het fijn zouden vinden om leerdoelen aan vragen te koppelen, opdat zij een betere terugkoppeling kunnen krijgen.

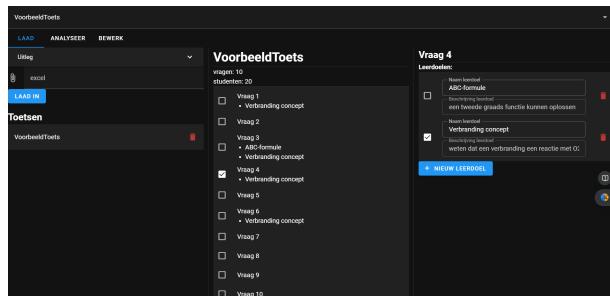


Figure 16: Voorbeeld inlaadpagina

In het **analyse scherm** moet voor een docent overzichtelijk weergegeven zijn welke vragen goed en minder goed gingen en de scores van de leerlingen. Hier moet ook te zien zijn welke vragen waarschijnlijk niet thuis horen in de toets, omdat de mensen met een hoog cijfer hem fout hebben, dit kan komen dat die leerlingen te ver doordenken en daardoor de vraag fout hebben. De vraag is of je die leerlingen wilt afstraffen voor het verder denken dan het juiste antwoord.

Het is ook mogelijk om hier opvallende correlaties tussen vragen weer te geven. Hier kan bijvoorbeeld worden laten zien dat mensen die vraag 5 fout hebben ook vraag 8 fout hebben. Dan is het mogelijk dat er 2x om dezelfde kennis wordt gevraagd, iets wat een toetsresultaat minder betrouwbaar maakt, omdat de stof dispropotioneel wordt getoetst (dit geldt ook voor een hoge correlatie tussen 2 juist gemaakte vragen).

Het kan ook mogelijk zijn om de correlaties tussen leerlingen te tonen om eventuele spiekers te vangen. Hierbij moet wel rekening gehouden worden met het feit dat 2 mensen met een hoog cijfer, waarschijnlijk beide dezelfde vragen goed en fout hebben. Deze correlatie wordt wel interessant bij bijvoorbeeld 2 6.5'en en precies dezelfde fouten.

Op het **bewerkscherm** kunnen bijvoorbeeld een aantal velden komen met de doelen van een docent. Bijvoorbeeld: een veld om het gewilde gemiddelde en de gewilde standaarddeviatie in te stellen. Daarmee berekent hij dat een nieuwe formule. Hier kan ook worden of een lineare of non-lineaire formule gebruikt wordt. Bij een non-lineaire formule behoudt iedereen met een onvoldoende zijn onvoldoende, maar zijn die onvoldoendes minder hoog. Hier kan ook op vraagniveau een scherm zijn om vragen eruit te gooien of half te laten meetellen, als blijkt dat ze wegnemen van de kennismeting van de toets.

B.4 Enquête

Eigenaar:	<i>Jonathan en Joost</i>
Doel(en):	• Inzicht krijgen in de mogelijkheid in de integratie van AI bij docenten op Het Ams
Subvragen:	• Hoe neem je een betrouwbare enquête? • Hoe zorg je ervoor dat mensen jouw enquête willen invullen?
Kader(s):	• TODO
Geschatte tijdkosten:	20 uur

4 Resultaten

A Inscannen

B Nakijken

C Analyseren

D Enquête

Vooraf het maken van ons model is aan alle docenten van Het Amsterdams Lyceum gevraagd wat hun gedachtes waren over het nakijken m.b.v. artificiële intelligentie. In de enquête werden diverse vragen gesteld om een breed beeld te krijgen van de standpunten en ervaringen van de docenten. Zo werd er gevraagd naar hun vakgebied en het aantal jaren dat ze al voor de klas staan. Ook werd geïnformeerd naar hun bekendheid met (generatieve) AI. Belangrijke thema's waren de redenen waarom docenten AI zouden willen gebruiken bij het nakijken van proefwerken en de zorgen die ze hierbij hebben. Daarnaast werd er gevraagd of zij denken dat AI zelfstandig toetsen kan nakijken en hoeveel leerlingen doorgaans hun beoordeling in twijfel trekken. Tot slot werd er ingegaan op de mogelijke invloed van AI op de relatie tussen docent en student.

Hoeveel jaar bent u al docent?	Ja	Een beetje	Nee	Total
1-5 jaar	4	4	0	8
5-10 jaar	5	4	0	9
Meer dan tien jaar	13	16	1	30
Minder dan één jaar	1	1	0	2
Grand Total	23	25	1	49

Table 3: Samenvatting van bekendheid met generatieve AI onder docenten.

Hoeveel jaar bent u docent?	Ja (%)	Nee (%)	Weet ik niet (%)	Total (%)
1-5 jaar	37.50	50.00	12.50	100.00
5-10 jaar	44.44	33.33	22.22	100.00
Meer dan tien jaar	23.33	40.00	36.67	100.00
Minder dan één jaar	50.00	50.00	0.00	100.00
Grand Total	28.57	40.82	30.61	100.00

Table 4: Percentageverdeling van vertrouwen met nakijken door AI onder docenten.

Hoeveel jaar bent u al docent?	Gemiddeld aantal leerlingen	Aantal docenten
1-5 jaar	3.375	8
5-10 jaar	1.222	9
Meer dan tien jaar	1.862	30
Minder dan één jaar	7.500	2
Grand Total	2.229	49

Table 5: Gemiddelde van het aantal leerlingen dat de beoordeling van docenten in twijfel trekt, per ervaringscategorie.

5 conclusie

- A Inscannen
- B Nakijken
- C Analyseren
- D Enquete

6 Discussie

- A Foutenanalyse

- A.1 Inscannen
- A.2 Nakijken
- A.3 Analyseren
- A.4 Enquête

- B vervolgonderzoek

7 Samenvatting onderzoek

8 Referenties

9 Bijlagen