

Academiejaar

2023 - 2024

# Large Language Models

## LLM Chatbots voor Domein Specifieke Vragen

**Vincent Verbergt**

Bachelorproef tot het behalen van een diploma Bachelor of Science in de  
**industriële wetenschappen: elektronica-ICT**

Promotoren

**Prof. dr. Peter Hellinckx**



Universiteit Antwerpen  
| Faculteit Toegepaste  
Ingenieurswetenschappen

# Inhoud

|   |                                          |    |
|---|------------------------------------------|----|
| 1 | Abstract .....                           | 3  |
| 2 | Inleiding.....                           | 3  |
| 3 | Modelkeuze .....                         | 4  |
|   | 3.1 Benchmarks selecteren.....           | 4  |
|   | 3.2 Model Analyse.....                   | 4  |
| 4 | Systeem evaluatie.....                   | 5  |
| 5 | Implementatie .....                      | 6  |
|   | 5.1 Finetuning .....                     | 6  |
|   | 5.2 Retrieval Augmented Generation ..... | 6  |
| 6 | Resultaten .....                         | 7  |
| 7 | Conclusies.....                          | 8  |
| 8 | Bibliografie .....                       | 9  |
| 9 | Appendix .....                           | 11 |
|   | 9.1 Tabellen.....                        | 11 |
|   | 9.2 Figuren .....                        | 14 |

# 1 Abstract

Deze publicatie omvat een vergelijkende studie van Large Language Modellen (LLM). Elk van deze modellen werd vergeleken op basis van veelgebruikte benchmarks en ook op de prestaties in een concrete use-case. Dit was een deelname aan een universitair examen. De methodes die getest werden waren finetuning en Retrieval Augmented Generation (RAG). We concluderen dat RAG de betere methode is, maar dat er zeer specifieke instellingen aan verbonden zijn. Finetuning is daarmee niet nutteloos, maar heeft andere use-cases waarin het diens nut kan bewijzen.

# 2 Inleiding

Sinds de publicatie van OpenAI's ChatGPT (GPT3.5) in het najaar van 2022, is er een explosieve stijging in het gebruik van artificiële intelligentie. Natural Language Processing (NLP) systemen, die snel menselijke teksten kunnen genereren (Generatieve AI), kunnen in veel velden de productiviteit verhogen. Deze systemen komen tegenwoordig ook vaak voor in helpdeskstelsystemen zoals Kate AI van KBC (KBC, n.d.). Voor programmeurs kunnen deze systemen grote bronnen documentatie doorspitten om zo behulpzame assistenten te zijn. Een dienst die dit adverteert is Kapa.ai, die Retrieval Augmented Generation (RAG) toepast in combinatie met het GPT-4 model (Kapa.ai, 2024).

Taalmodellen moeten gigantische hoeveelheden data verwerken in hun trainingsperiode. De data waarop getraind wordt, zijn tekstuele internetbronnen zoals Wikipedia. Modellen zoals GPT-4 gebruiken de transformer architectuur om dit proces aanzienlijk te versnellen (Ashish Vaswani, 2023). Hierdoor bevatten deze een solide basiskennis in nagenoeg alle domeinen. Maar wanneer een expert vragen over diens domein vraagt, zal deze concluderen dat de antwoorden tekortschieten. Het is duidelijk dat deze modellen *Jacks of all trades* zijn, *but masters of none*. Kunnen deze modellen toch aangepast worden om dit te verhelpen?

## “Hebben de LLM’s het in zich om expertise te vergaren in een specifiek domein?”

RAG is een methode die toelaat modellen gaandeweg kennis te laten vergaren. Deze techniek vergelijkt de vraag van de gebruiker eerst met de kennisbron. De meest relevante passages worden uit de kennisbron gehaald en meegegeven aan een LLM. Deze genereert op basis van de vraag en de meegegeven passages een kort, natuurlijk antwoord. (Patrick Lewis, 2021)

RAG is zeker niet de enige manier voor LLM's om kennis te verkrijgen. Als we de prominente GPT-4 een vraag stellen, is het duidelijk dat deze een brede basiskennis heeft. Deze kennis werd opgedaan tijdens de trainingsfase van het model. Dit vergt wel gigantisch veel data. Als een model getraind wordt op een te kleine kennisbron, vergaat deze geen taalinzicht. Wel kan een voorgetraind model een korte secundaire training op de domein specifieke kennisbron doen. Dit heet finetuning (Jeong, 2024).

Beide methodes hebben hun voor- en nadelen. Het verwachte resultaat is dat een gefinetuned model kleiner is en daardoor een snellere inferentietijd heeft, maar de antwoorden meer fouten zullen bevatten. RAG vereist een grote attentiespanne/context-window. Bij deze methode moeten ook de passages uit de kennisbron verwerkt worden door het model. Dit heeft als nadeel dat het systeem geen antwoord kan genereren tijdens de verwerkingsperiode. Hierdoor is de kennis wel up-to-date en kunnen complexere modellen ingezet worden waardoor betere antwoorden gevormd kunnen worden. (Angels Balaguer, 2024)

Om de resultaten op het vlak van kennisvergaring in een specifiek domein effectief te vergelijken, hebben we een eigen benchmark opgezet . Deze benchmark test niet enkel de kennis van het systeem, maar ook diens vaardigheden als assistent. Dit houdt in dat het systeem correct kan inschatten of de kennisbron voldoende

informatie over de vraag bevat. Ook moet het hiermee dus vragen buiten het expertisedomein negeren. Daarbovenop wordt ook de redeneringsvaardigheid van de systemen getest: kunnen ze informatie combineren om tot een nieuwe waarheid te komen.

## 3 Modelkeuze

Dit onderzoek heeft drie doelstellingen:

1. Nagaan of een LLM expertise kan verwerven in een specifiek domein met behulp van een kennisbron.
2. Nagaan wat de meest relevante methodes hiervoor zijn en hoe goed deze presteren.
3. Nagaan welke methodes waar toegepast kunnen/mogen worden.

### 3.1 Benchmarks selecteren

De finetuning en RAG methodes vereisen beiden reeds getrainde taalmodellen. Het is belangrijk om een sterk model als basis te kiezen. Op voorhand werd beslist welke eigenschappen belangrijk zijn voor de methodes. Daarna zochten we verschillende testen/benchmarks die op deze eigenschappen testen. Via deze methode werd er een kwantitatief optimaal model geselecteerd.

We hebben ervoor gekozen om modellen vooral een score te geven op hun taalcomprehensie ongeacht het toepassingsdomein waarvoor ze werden ontwikkeld. De state-of-the-art testen in deze context zijn de HellaSwag- en Winogrande benchmark. De benchmarks zijn zo opgesteld dat ze voor mensen zeer makkelijk zijn, maar voor statistische taalmodellen zeer complex.

### 3.2 Model Analyse

Omdat we het volledige spectrum van modellen wilden vergelijken, was het belangrijk om te differentiëren tussen open- en closed-source modellen. Omdat open-source de volledige structuur van de modellen vrijgeeft, zijn deze makkelijk te finetunen. In de closed-source wereld moet de maker dit explicet toelaten. Daarnaast is het ook onmogelijk om deze closed-source modellen zelf te hosten. Alle interacties met de taalmodellen gaan via een betalende API. De beperkingen die gepaard gaan met closed-source modellen zijn te wijten aan hun grotere omvang waardoor hun hardware kosten snel oplopen.

In de open-source LLM wereld is HuggingFace het meest gebruikte platform om modellen te delen (Hugging Face, n.d.). Ze voorzien een extensief leaderboard waar de HellaSwag en Winogrande scores ook beschikbaar zijn (Edward Beeching, 2023). Een eerste analyse die gemaakt werd, gaf inzicht in de beschikbare modelgroottes met behulp van een frequentieplot. In Figuur 1 konden we 3 groepen vormen. De lijnen die deze groepen verdelen, lagen op 25- en 58 miljard parameters.

Hierna werden per groep de modellen gerangschikt op de gemiddelde score op de HellaSwag en Winogrande benchmark. Deze benchmarks evalueren duidelijk vergelijkbare eigenschappen waardoor er een correlatie is in hun testscores. De resultaten zijn terug te vinden in Tabel 1, Tabel 2 en Tabel 3.

In de resultaten valt op dat Mixtral de winnaar was voor de kleinere modellen (< 58 miljard parameters). Aan de andere kant was Llama de winnaar bij de grotere modellen. Ook zien we dat over de drie groepen heen de scores zeer dicht bij elkaar lagen. Het verschil in de gemiddeldes van de best presterende modellen in elke categorie was kleiner dan 5%.

Omdat finetuning een zeer intensief proces is, was gekozen om een klein, performant model te gebruiken zodat de totale rekenkundige complexiteit niet zou oplopen. De verschillende architecturen werden geanalyseerd op hun scores en gemiddelde grootte (Tabel 4). De gelimiteerde grootte in combinatie met hoge scores maakte de Mistral modellen de beste kandidaat voor finetuning.

Voor de closed-source modellen werd een gelijkaardige analyse gedaan (Tabel 5). Aangezien we de modellen niet zelf moesten hosten, was de grootte aanzienlijk minder belangrijk. Uit de data bleek dat recente Claude-3 modellen van Anthropic zeer hoog scoren (Anthropic, 2024). Google's Gemini modellen liepen merkbaar achter op de concurrentie. Ook bleken deze nog niet beschikbaar te zijn in Europa waardoor ze buiten het onderzoek vielen. Voor dit onderzoek gebruikten we de OpenAI- en Anthropic modellen.

## 4 Systeem evaluatie

Om de verschillende systemen met elkaar te kunnen vergelijken, hebben we een eigen benchmark geïntroduceerd. De beste testmethodiek om veel resultaten uit NLP systemen te kunnen vergelijken, is het gebruik van een meerkeuzetest (Liwen Zhang, 2023). Het is een methode die de analyse- en synthesevaardigheid van een systeem test. Deze kwantitatieve aanpak maakt dat een programma deze testen automatisch kan verbeteren. Hierdoor kunnen we veel testen uitvoeren. Daartegenover staat wel dat deze methode weinig ruimte voor nuance overlaat.

Uit onderzoek bleek dat, bij meerkeuzetests, een redenering bij het antwoord vereisen, leidde tot betere resultaten. Deze redeneringsstap wordt de Chain-of-Thought (CoT) genoemd. Daarbovenop kan deze methodiek ervoor zorgen dat modellen deductievaardigheden vertonen. De testbank gebruikte deze CoT methode zodat de systemen beter zouden presteren. (Jason Wei, 2022)

Een keuze die gemaakt wordt bij het testen van een NLP systeem is of er voorbeeldvragen én antwoorden meegegeven worden. Het meegeven van voorbeeldvragen wordt N-shotting genoemd waarbij N de hoeveelheid voorbeelden is. Veelvoorkomende waardes voor N zijn 0, 5, 10. Afhankelijk van de toepassing en het model is er een optimale waarde N waarvoor de score het hoogst ligt. Voor onze keuze van N werd gekeken naar het applicatiedomein van het finale systeem. Dit is een chatbot waar de assistent maar één kans heeft om te antwoorden. Hierdoor werd gekozen voor een zero-shot omgeving.

De vragen waaruit de zero-shot CoT meerkeuze benchmark bestond, moeten representatief zijn voor de eigenschappen die we willen evalueren. De testbank moest kennis, redenering, hallucinatieresistentie en domeingerichtheid testen. De verschillende vraagcategorieën staan beschreven in Tabel 6. Elke vraag bevat vijf opties waar het systeem uit kan kiezen. Drie van deze opties zijn mogelijke antwoorden op de gestelde vraag. Optie vier dient voor wanneer geen enkel antwoord juist is. De laatste optie geeft aan dat de vraag niet binnen het domein ligt.

De specifieke casestudie die we testen, is expertise in het Digital Image Processing (DIP) domein. DIP heeft een zeer wiskundige aard waarvan geweten is dat dit lastig kan zijn voor LLM's (Janice Ahn, 2024). Anderzijds kunnen de processen die de afbeeldingen ondergaan kwalitatief besproken worden. Hiermee hopen we een mooie balans te maken tussen wiskunde en beeldende taal. Alle vragen komen uit de cursus Digital Image Processing die aangeboden wordt door de universiteit van Antwerpen. Alle onderdelen van de cursus zijn terug te vinden in de vragen. (Daems, 2023)

Elk systeem wordt 1, 3 of 5 keer getest afhankelijk van de financiële kosten die gepaard gaan met het uitvoeren van een test. De systemen zijn niet deterministisch en vertonen daarom willekeurig gedrag. Om de variantie en nuances van de best presterende systemen in kaart te brengen wordt ook nog eens getest met open vragen die manueel verbeterd worden.

## 5 Implementatie

### 5.1 Finetuning

Finetunen en trainen van LLM's is een zeer hardware intensieve taak. De voornaamste bottleneck was de geheugenvereiste van de modellen. Om voor een model te kunnen bepalen hoeveel geheugen het in beslag zou nemen, werd volgende formule gebruikt:

$$\text{Totaal geheugen} = \#Parameters * \frac{\#Bytes}{Parameter}$$

Bij het trainen van een transformer model bleek dat de effectieve vereisten viermaal groter waren dan het model zelf. Voor het 14GB Mistral model resulteerde dit in een 56GB geheugenverbruik. Om dit te halen werd er getraind met behulp van cloud-resources. Hiervoor werd HuggingFace's AutoTrain gebruikt (Auto Train, n.d.). Dit is een relatief goedkope manier om modellen te finetunen a.d.h.v. datasets.

Nadat het model getraind was, werd het gedownload en ingeladen. Om de inferentietijd te verkorten werd het model gekwantificeerd naar een vierde van de originele grootte. Het ingeladen model werd in een pipeline gestoken. Deze bevatte nog enkele extra prompts om het domein aan te geven en het meerkeuze testformaat te schetsen. De pipeline werd achteraf in onze automatische benchmark omgeving gestoken.

### 5.2 Retrieval Augmented Generation

Het doel van RAG was om de kennisbron te gaan doorzoeken op passages die relevant zijn aan de gestelde vraag (Patrick Lewis, 2021). De methode die geïmplementeerd werd, was Dense Passage Retrieval (DPR). DPR is een *dual encoder* framework dat de semantische betekenis van de vraag en context vastlegt in een hoog dimensionale vector. Vervolgens berekent het systeem de cosinus afstand tussen de vraag- en contextvectoren. Hoe kleiner deze afstand, hoe relevanter de passage aan de vraag is. Met behulp van een Faiss-index werd dit proces aanzienlijk versneld (Matthijs Douze, 2024). (Vladimir Karpukhin, 2020)

De passage kon semantisch gelijkend zijn aan de vraag, maar het antwoord erop niet bevatten. Hiervoor werden twee strategieën uitgewerkt. Bij de eerste werd de alinea waar de zin in voorkwam, meegegeven. Bij de andere werd de volledige pagina meegegeven. Er werd ingespeeld op het feit dat de relevante informatie in de buurt stond van de relevante zinnen. Bij de pagina strategie was er een grotere kans dat de passage het antwoord bevatte, maar dit ten koste van de hoeveelheid passages dat er meegegeven konden worden. Daartegenover werd er bij de alinea strategie meerdere kleinere passages ingevoegd waarvan de meeste geen zinvolle extra informatie bevatten.

DPR bleek een handige techniek te zijn om semantisch een kennisbron te doorzoeken (Vladimir Karpukhin, 2020). Het grote nadeel van deze methode is dat de initiële berekeningen zeer intensief zijn. Wanneer een kennisbron hoog volatiel is, zijn andere zoekmethodes zoals *N-gram similarity search* beter (Davide Buscaldi, 2013).

Het finale systeem had de volgende vorm. Op voorhand werd de contextvector voor elke zin berekend met behulp van de context encoder. Hierna werd de Faiss-index voor alle contextvectoren berekend. Wanneer een vraag gesteld wordt, berekenen we hier de vraagvector voor. Dan wordt de dichtstbijzijnde contextvector opgezocht. Hierna kijken we op welke pagina of in welke alinea deze zin voorkomt. Deze passage wordt dan meegegeven in de prompt. Hierna gebruiken we de tweede dichtstbijzijnde contextvector en diens passage om de prompt aan te vullen. Dit wordt herhaald tot de volledige context-window gevuld is.

## 6 Resultaten

We hebben drie grote testen gedaan met de benchmark. De eerste test was op de lokale modellen, deze waren het goedkoopst om te testen. Deze doorzocht dus de parameterruimte voor optimale strategieën. Daarna gingen we met deze kennis GPT-3.5-Turbo testen om de optimale RAG-strategie te vinden. Deze strategie werd dan gebruikt om de prestaties van de geselecteerde closed-source modellen te meten.

Tijdens de lokale testen werden de volgende modellen en systemen getest: Mistral met en zonder finetuning en RAG, Mixtral met en zonder RAG en als laatste Llama-2 zonder RAG (de hardware kon de verhoogde context-window niet aan). Deze modellen werden getest op de volledige meerkeuze benchmark. Elk model dat met RAG getest werd, lieten we de pagina en alinea strategie gebruiken, beide met een 2048- en 4096 woorden context-window. De resultaten van deze testen zijn terug te vinden in Figuur 2, Figuur 3, Figuur 4 en Figuur 5.

Uit de data blijkt dat de Mixtral architectuur het beste presteert, maar dit ten koste van een hoge inferentietijd. Ook valt op te merken dat er maar één RAG-strategie de scores verbetert, namelijk een context-window van 2048 woorden met pagina level retrieval (Page2k). Ten laatste lijkt de finetuning methode de scores enkel maar te laten dalen. In het algemeen scoren de grotere modellen zeer goed op de test.

De page2k strategie bleek niet enkel het beste te presteren bij de lokale modellen, maar ook bij GPT-3.5. Dit is te zien in Figuur 6. Hier valt ook weer op dat het basismodel zeer hoog presteert. We kunnen afleiden dat er een lineaire correlatie is tussen context-grootte en (Figuur 7). Dit verband viel veel moeilijker af te leiden uit Figuur 5 waar Mixtral hier vooral van afweek, dit is mogelijk te wijten aan de *sparse mixture of experts* model (Mistral AI team, 2023).

Met de optimale page2k strategie evaluateerden we nu de closed-source modellen. Hiervoor gebruikten we de modellen uit onze closed-source preselectie: OpenAI en Anthropic. Van OpenAI kozen we voor GPT-3.5-Turbo en GPT-4-Turbo. Bij Anthropic werden de Claude-3 modellen geselecteerd. Elk model onderging 3 bevragingen met de optimale RAG strategie. De resultaten hiervan zijn te vinden in Figuur 8, Figuur 9.

Uit deze data blijkt dat de Claude-3 modellen ontzettend hoog presteren. Hierbij is het kleinste model, Haiku (met de laagste inferentietijd), de hoogste scorende LLM. Daarbovenop is het 60 keer goedkoper dan het Opus model, 40 keer goedkoper dan GPT-4-Turbo en 2 keer goedkoper dan GPT-3.5 (Anthropic, n.d.), (OpenAI, n.d.). De reden dat de Claude-3 modellen hoger scoorden, valt te wijten aan hun nieuwere architectuur.

In een allerlaatste stap werden 4 modellen nogmaals getest met 5 open vragen. Deze vragen werden opgesteld in samenwerking met de DIP cursusleider, Walter Daems. De antwoorden werden achteraf handmatig verbeterd. De 4 geselecteerde modellen die werden getest waren:

1. het beste closed-source model Haiku
2. het meest bekende closed-source model GPT-4-Turbo
3. het beste open-source model Mixtral
4. het enige gefinetuned model, Mistral.

Elk van deze modellen onderging de test drie keer. Achteraf werd de minimumscores berekend door altijd de slechtste antwoorden over de drie testen te kiezen, hetzelfde werd gedaan voor de maximumscores. Deze verschillen van de minimum- en maximum testscores omdat deze het minimum en maximum zijn van een volledige test. Ook werd een gemiddelde score berekend. De resultaten hiervan zijn te vinden in Tabel 7.

Hier zien we dat het Haiku model nog steeds goed presteert, maar wel hoge variantie heeft in de kwaliteit van de antwoorden. We vermoeden dat de zeer hoge score in de meerkeuze benchmark veroorzaakt werd door het

systeem dat patronen in de vraagstelling herkende in plaats van na te denken over een antwoord. We zien wel dat GPT-4-Turbo zeer hoog presteert over de hele lijn en in alle opzichten het minste variatie vertoont.

## 7 Conclusies

Uit de resultaten kunnen we afleiden dat de modellen enorm veel basiskennis bezitten. Dit zien we aan de hoge baselinescores. Daarbij zien we dat een groter model gemiddeld hoger scoort. Hiertegenover staat dan wel dat een groter model een hogere inferentietijd heeft en meer geheugen in beslag neemt waardoor de kost hiervan ook hoger komt te liggen.

Finetuning blijkt een slechte methode te zijn om modellen kennis te doen vergaren. We vermoeden daarom dat finetuning een andere use-case heeft, namelijk vaardigheden leren. Zo zien we dat veel modellen op HuggingFace een instructie variant hebben (Hugging Face, n.d.). Deze varianten zijn gefinetuned zodat deze kunnen omgaan met specifieke instructieformaten en chat templates. Dit kan hen een effectievere assistent maken.

Voor RAG systemen moet opgelet worden dat een optimale strategie gebruikt. Wanneer we dit niet zouden doen, kan de kwaliteit van de antwoorden zelfs dalen. We vermoeden dat dit fenomeen veroorzaakt wordt door attentiemechanismes die een slechte invloed gaan uitoefenen bij te lange prompts. Ook zorgt de langere prompt ervoor dat het model de vraag oppervlakkiger leest en daarmee minder correct antwoord. Dit komt omdat de vraag nu maar een klein deel van de volledige prompt is.

Wanneer de antwoorden op de vragen van dichterbij geëvalueerd werden, viel op dat de kwaliteit hoog lag. Er werd nooit een volledig fout antwoord gegeven. Vaak moesten we punten aftrekken omdat het voorgestelde antwoord niet uit de kennisbron kwam. Ook zien we dat er een hoge variantie is op de antwoorden omwille van het niet deterministische gedrag van de systemen. Vaak wanneer een model fout antwoord, was deze er volledig van overtuigd dat het correct was.

Een optimaal systeem voor een kennisbron te bevragen, is een LLM met veel parameters. Dit model bevindt zich in een RAG omgeving waarbij veel getest moet worden wat de optimale strategie en context-window is. De *retrieval* van de relevante passage gebeurt best met behulp van een semantische *similarity search*. Dit werkt het beste voor statische kennisbronnen. Als de mogelijkheid er is, kan het model best nog gefinetuned worden om beter te presteren met lange RAG prompts. Ook kan er op het einde nog gefinetuned worden om het als instructiemodel te laten dienen.

## 8 Bibliografie

- Angels Balaguer, V. B. (2024, Januari 30). *RAG vs Fine-tuning: Pipelines, Tradeoffs, and a Case Study on Agriculture*. Opgeroepen op Maart 10, 2024, van Arxiv: <https://arxiv.org/abs/2401.08406>
- Anthropic. (2024, Maart 4). *The Claude 3 Model Family: Opus, Sonnet, Haiku*. Opgeroepen op April 27, 2024, van Anthropic: [https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Model\\_Card\\_Claude\\_3.pdf](https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Model_Card_Claude_3.pdf)
- Anthropic. (sd). *Claude 3 Pricing*. Opgeroepen op Mei 10, 2024, van Anthropic: <https://www.anthropic.com/api>
- Ashish Vaswani, N. S. (2023, Augustus 2023). *Attention Is All You Need*. Opgeroepen op Februari 13, 2024, van Arxiv: <https://arxiv.org/abs/1706.03762>
- Auto Train. (sd). *Create powerful AI models without code*. Opgeroepen op April 17, 2024, van Hugging Face: <https://huggingface.co/autotrain>
- Daems, W. (2023). *Digital Image Processing*. Antwerpen: University of Antwerp.
- Davide Buscaldi, J. L. (2013, Juni 1). *LIPN-CORE: Semantic Text Similarity using n-grams, WordNet, Syntactic*. Opgeroepen op Mei 1, 2024, van Hal Open Science: <https://hal.science/hal-00825054/>
- Edward Beeching, C. F. (2023). *Open LLM Leaderboard*. Opgeroepen op Maart 15, 2024, van Hugging Face: [https://huggingface.co/spaces/HuggingFaceH4/open\\_llm\\_leaderboard](https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard)
- Hugging Face. (sd). Opgeroepen op Maart 16, 2024, van Hugging Face: <https://huggingface.co/>
- Janice Ahn, R. V. (2024, Januari 31). *Large Language Models for Mathematical Reasoning: Progresses and Challenges*. Opgeroepen op Februari 28, 2024, van Arxiv: <https://arxiv.org/abs/2402.00157v1>
- Jason Wei, X. W. (2022, Oktober 31). *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*. Opgehaald van Open Review: [https://openreview.net/forum?id=\\_VjQIMeSB\\_J](https://openreview.net/forum?id=_VjQIMeSB_J)
- Jeong, C. (2024, Januari 24). *Fine-tuning and Utilization Methods of Domain-specific LLMs*. Opgeroepen op Maart 3, 2024, van Arxiv: <https://arxiv.org/abs/2401.02981>
- Kapa.ai. (2024). *How Does Kapa Work?*  Opgeroepen op Februari 13, 2024, van Kapa.ai: <https://docs.kapa.ai/#how-does-kapa-work->
- KBC. (sd). *Maak kennis met Kate*. Opgeroepen op Februari 13, 2024, van KBC: <https://www.kbc.be/ondernemen/nl/nieuws/provibes/algemeen/kate.html#:~:text=Kate%20is%20een%20digitale%20assistent,Live%2C%20die%20je%20verder%20helpt>
- Liwen Zhang, W. C. (2023, Augustus 19). *FinEval: A Chinese Financial Domain Knowledge Evaluation Benchmark for Large Language Models*. Opgeroepen op Maart 15, 2024, van Arxiv: <https://arxiv.org/abs/2308.09975>
- Matthijs Douze, A. G.-E. (2024, Januari 16). *The Faiss library*. Opgeroepen op April 9, 2024, van Arxiv: <https://arxiv.org/abs/2401.08281>

Mistral AI team. (2023, December 11). *Mixtral of experts*. Opgeroepen op April 11, 2024, van MistralAI: <https://mistral.ai/news/mixtral-of-experts/>

OpenAI. (sd). *Pricing*. Opgeroepen op Mei 10, 2024, van OpenAI: <https://openai.com/api/pricing/>

Patrick Lewis, E. P.-t. (2021, April 12). *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*. Opgeroepen op Februari 27, 2024, van Arxiv: <https://arxiv.org/abs/2005.11401>

Vladimir Karpukhin, B. O.-t. (2020, September 30). *Dense Passage Retrieval for Open-Domain Question Answering*. Opgeroepen op April 9, 2024, van Arxiv: <https://arxiv.org/abs/2004.04906>

## 9 Appendix

### 9.1 Tabellen

| Architectuur       | HellaSwag (%) | Winogrande (%) | Model                                                  |
|--------------------|---------------|----------------|--------------------------------------------------------|
| MixtralForCausalLM | 89.02         | 89.27          | zhengr/MixTAO-7Bx2-MoE-Instruct-v6.0                   |
| MixtralForCausalLM | 89.72         | 87.85          | rizla/rizla-17                                         |
| MixtralForCausalLM | 89.3          | 88.24          | yunconglong/Truthful_DPO_TomGrc_FusionNet_7Bx2_MoE_13B |

Tabel 1 Top 3 scores modellen [0, 25[ miljard parameters.

| Architectuur       | HellaSwag (%) | Winogrande (%) | Model                           |
|--------------------|---------------|----------------|---------------------------------|
| CohereForCausalLM  | 87.96         | 83.82          | CohereForAI/c4ai-command-r-plus |
| MixtralForCausalLM | 87.73         | 82.56          | Swisslex/Mixtral-8x7b-DPO-v0.2  |
| MixtralForCausalLM | 87.13         | 82.95          | Steelskull/Lumosia-MoE-4x10.7   |

Tabel 2 Top 3 scores modellen [25, 58[ miljard parameters.

| Architectuur     | HellaSwag (%) | Winogrande (%) | Model                      |
|------------------|---------------|----------------|----------------------------|
| LlamaForCausalLM | 89.77         | 87.53          | MTSAIR/MultiVerse_70B      |
| LlamaForCausalLM | 88.6          | 85.4           | Undi95/Miqu-70B-Alpaca-DPO |
| LlamaForCausalLM | 88.61         | 85.32          | 152334H/miqu-1-70b-sf      |

Tabel 3 Top 3 scores modellen 58+ miljard parameters.

| Architectuur       | Gemiddelde (%) | #Params (Miljard) |
|--------------------|----------------|-------------------|
| MixtralForCausalLM | 87.20          | 24.15             |
| CohereForCausalLM  | 86.98          | 45.015            |
| MistralForCausalLM | 85.46          | 7.24              |

Tabel 4 Top 3 scores per architectuur.

| Model            | HellaSwag (%) | WinoGrande (%) |
|------------------|---------------|----------------|
| GPT-4            | 95.3          | 87.5           |
| GPT-3.5          | 85.5          | 81.6           |
| Claude-3 Opus    | 95.4          | 88.5           |
| Claude-3 Sonnet  | 89.0          | 75.1           |
| Claude-3 Haiku   | 85.9          | 74.2           |
| Gemini 1.0 Ultra | 87.8          | /              |
| Gemini 1.5 Pro   | 92.5          | /              |
| Gemini 1.0 Pro   | 84.5          | /              |

Tabel 5 Scores closed-source modellen (Anthropic, 2024) .

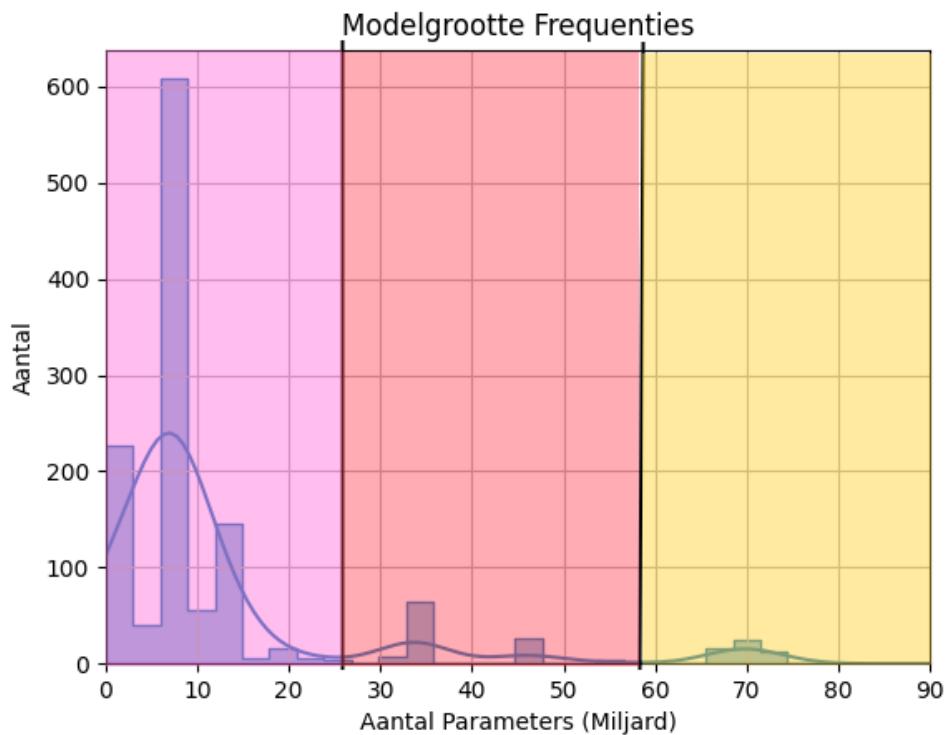
| Categorie     | Beschrijving                                                                    | Test voor...            | Hoeveelheid Vragen |
|---------------|---------------------------------------------------------------------------------|-------------------------|--------------------|
| Letterlijk    | Vragen die beantwoord konden worden met letterlijke passages uit de kennisbron. | Kennis                  | 13                 |
| Deductie      | Vragen die een extra denkstap vereisten.                                        | Redeneringsvaardigheid  | 3                  |
| Fout          | Vragen waar geen juist antwoord aanwezig was.                                   | Hallucinatieresistentie | 4                  |
| Buiten Domein | Vragen die niet relevant aan het domein zijn.                                   | Domeingerichtheid       | 3                  |

Tabel 6 Verdeling vragen benchmark

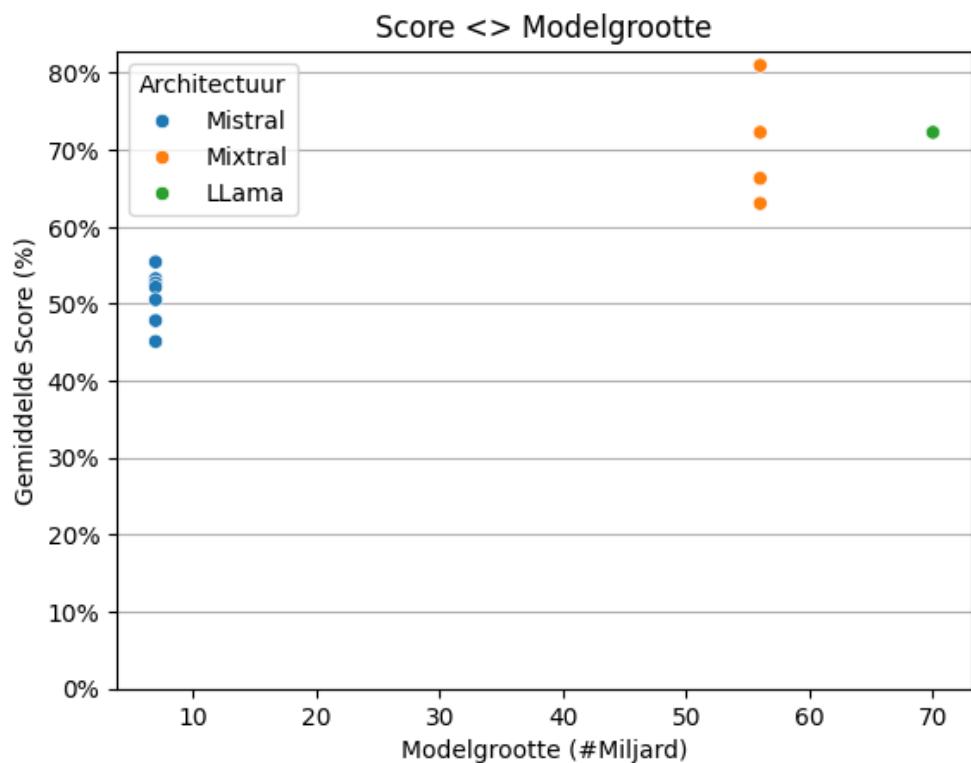
| Model Naam   | Minimum Score | Maximum Score | Gemiddelde Score | Minimum Test Score | Maximum Test Score |
|--------------|---------------|---------------|------------------|--------------------|--------------------|
| GPT-4-Turbo  | 7             | 9.25          | 8.25             | 7.75               | 9                  |
| Haiku        | 4.5           | 9             | 7                | 5.5                | 8                  |
| Mistral-FT   | 2.75          | 6             | 4.5              | 3                  | 5.5                |
| Mixtral-8x7B | 2.75          | 7             | 4.5              | 3                  | 5.75               |

Tabel 7 Scores open vragen test (/10).

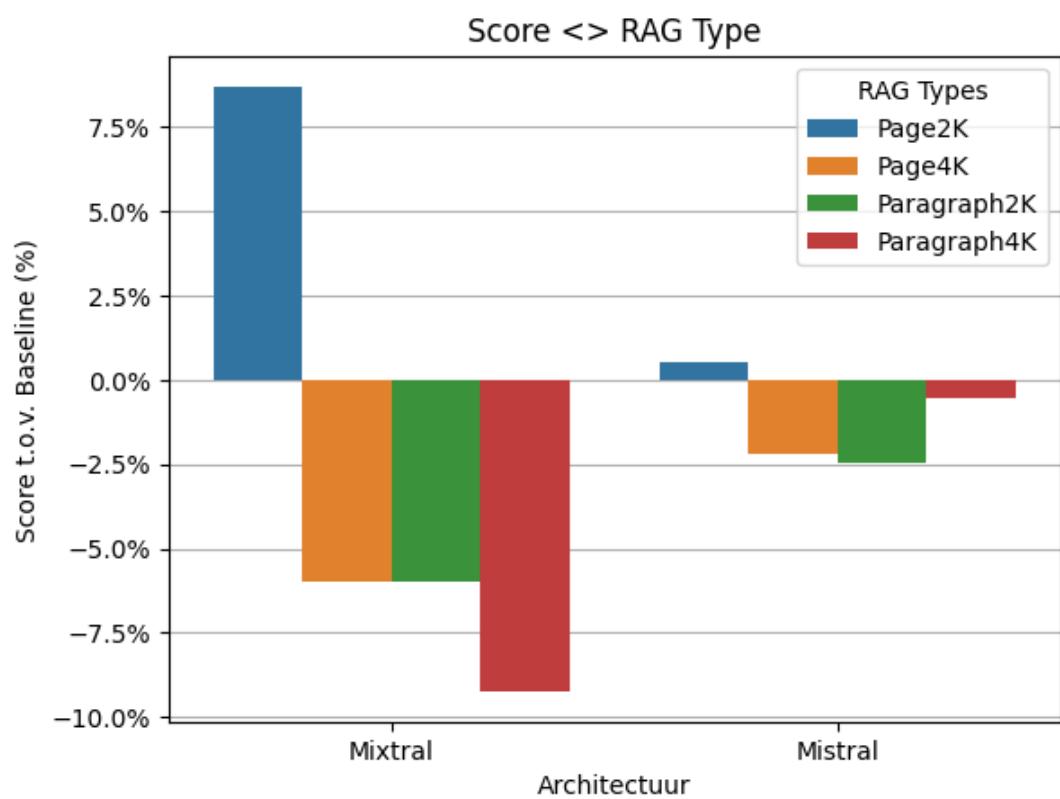
## 9.2 Figuren



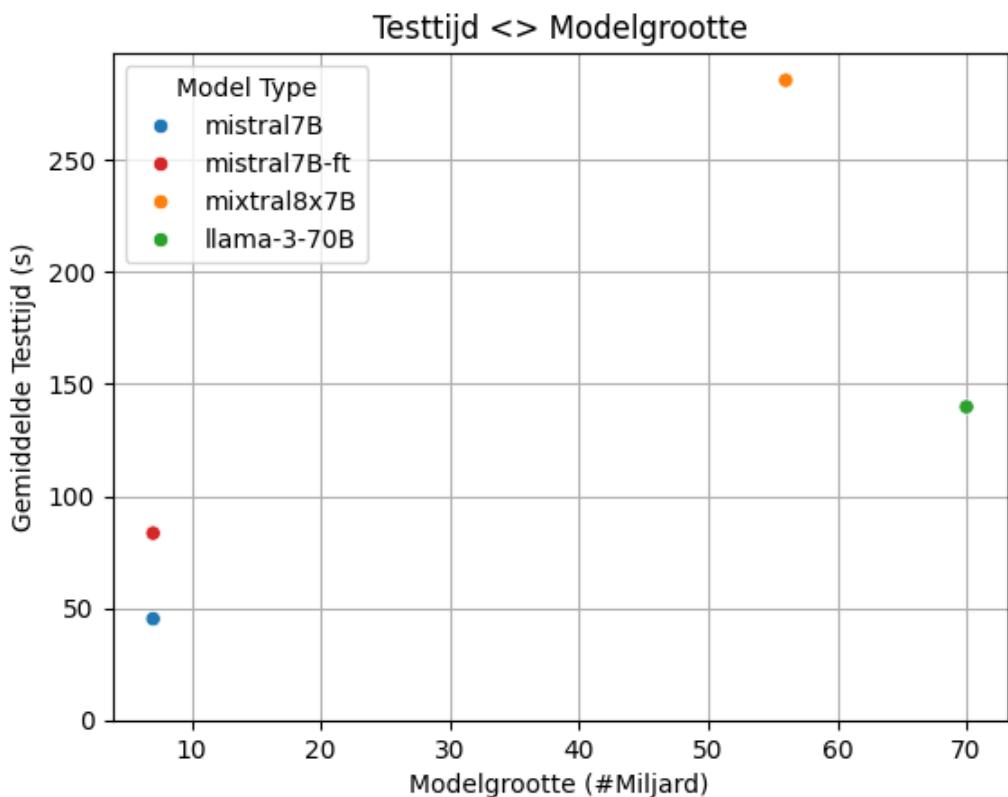
Figuur 1 Modelgrootte frequenties verdeeld in 3 groepen.



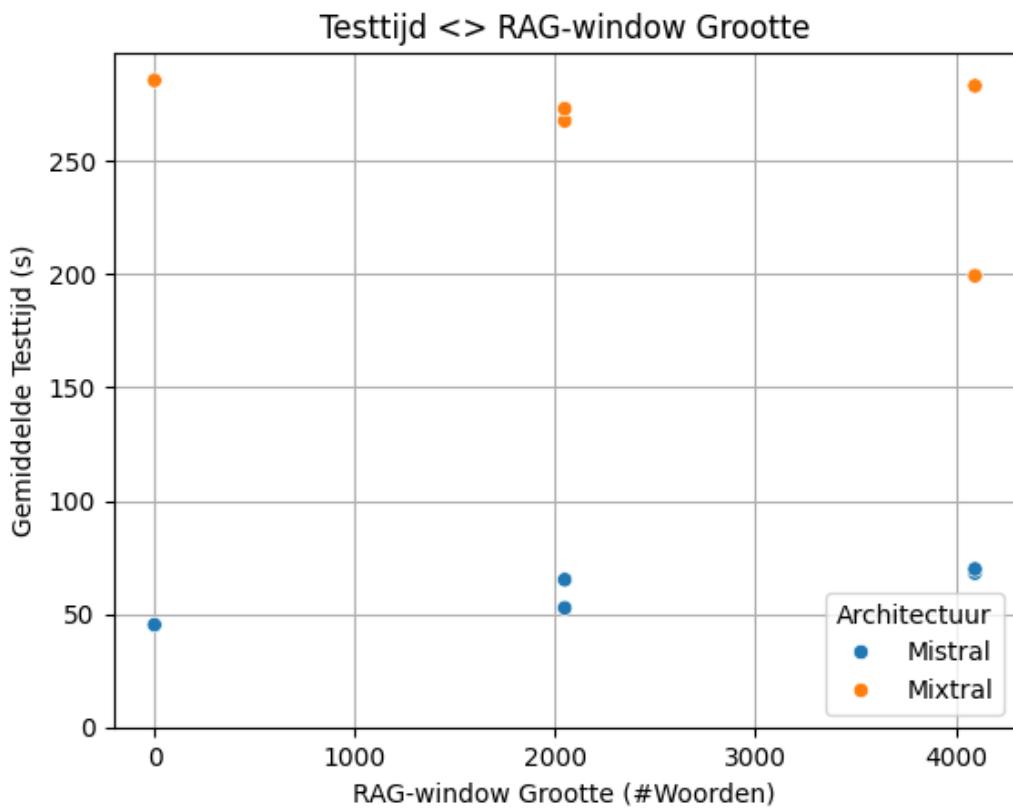
Figuur 2 [Lokaal] Score t.o.v. architectuur en modelgrootte.



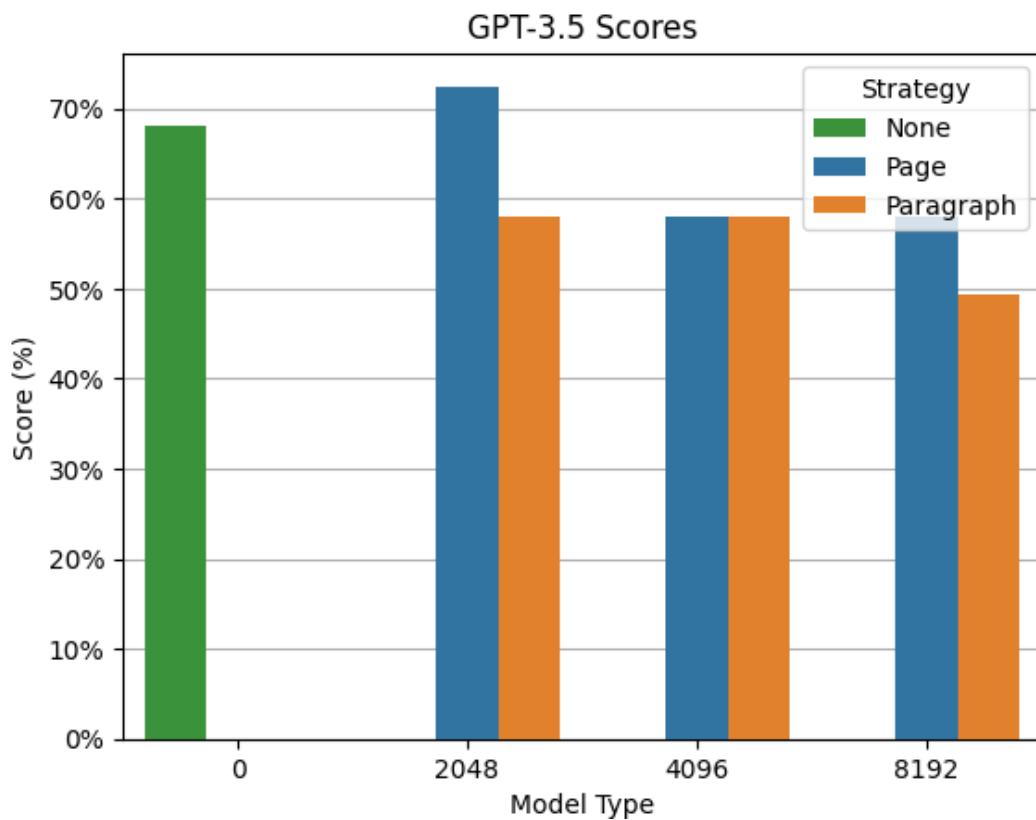
Figuur 3 [Lokaal] Scores t.o.v. de basis prestaties voor verschillende RAG strategieën.



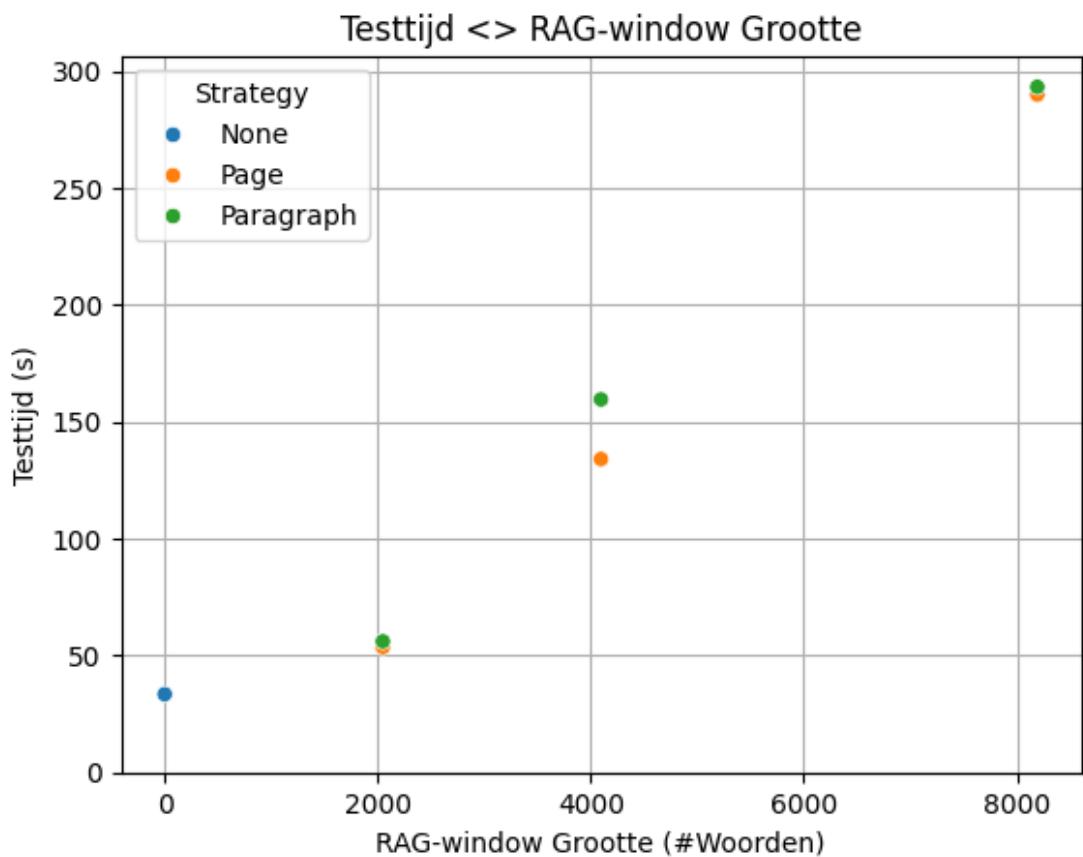
Figuur 4 [Lokaal] Gemiddelde testtijd voor verschillende modelgroottest.



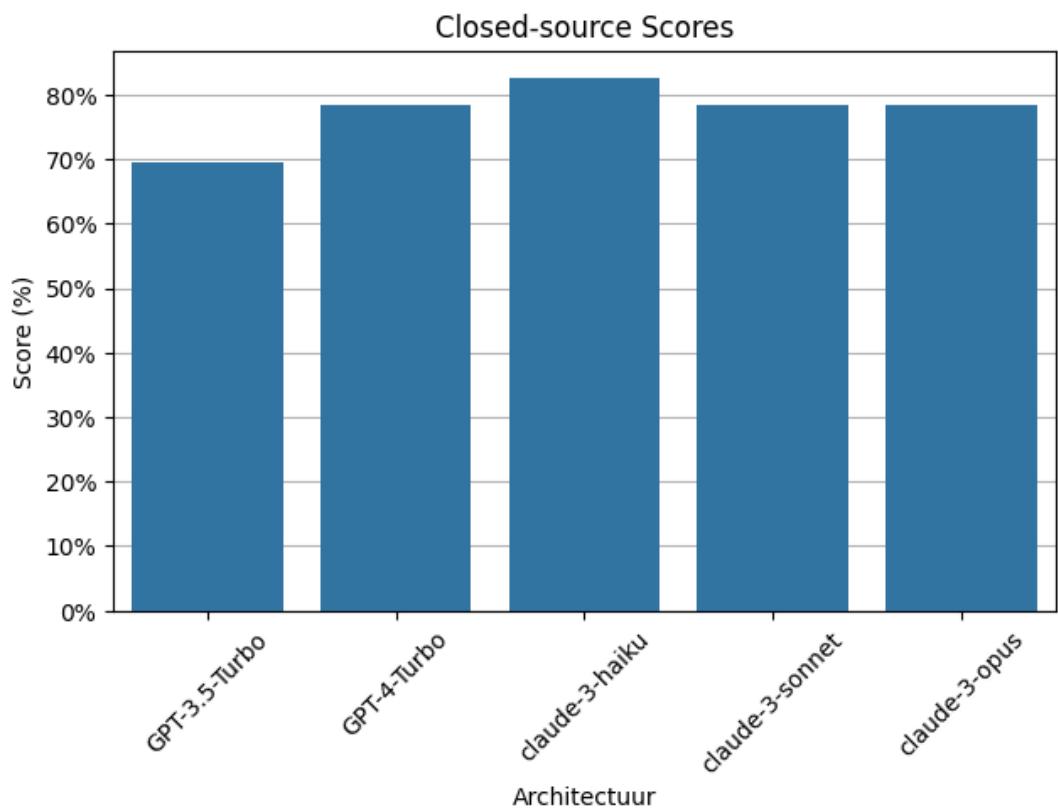
Figuur 5 [Lokaal] De invloed van de RAG-window grootte op de gemiddelde testtijd.



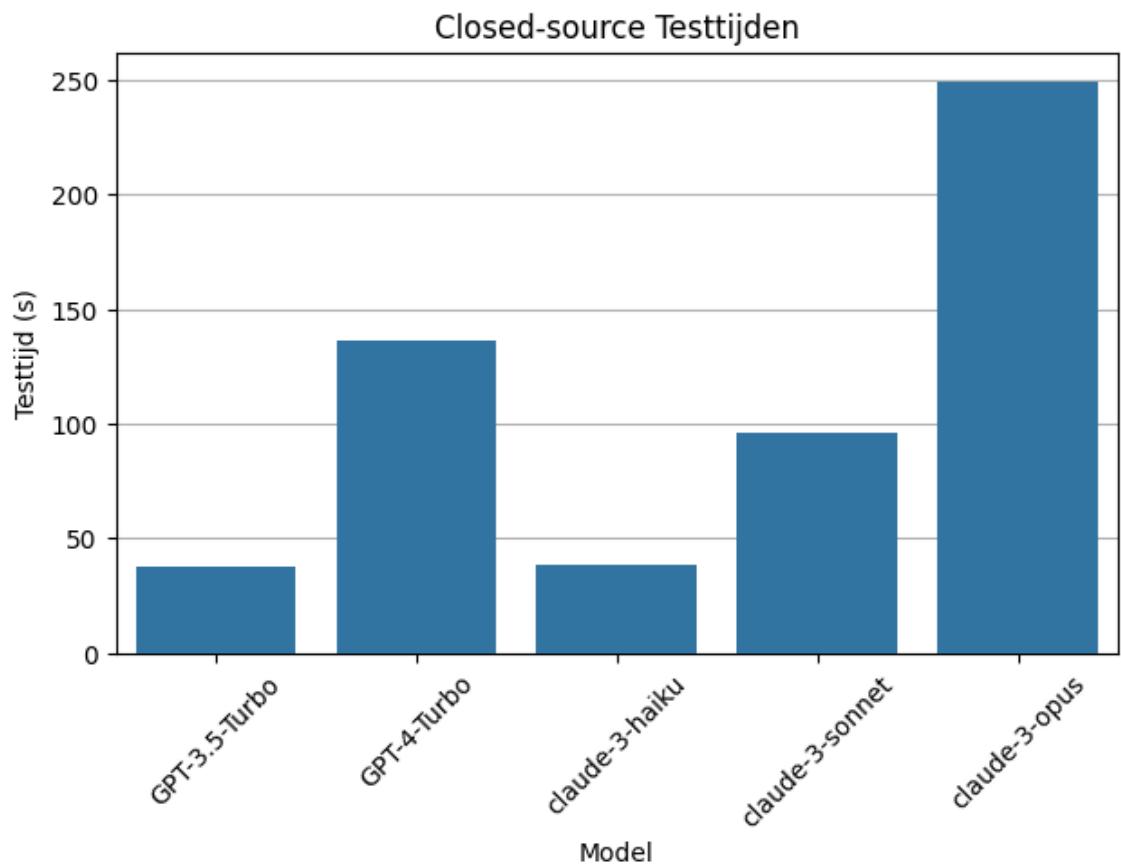
Figuur 6 [GPT-3.5] De scores voor verschillende RAG-window groottes en strategieën.



Figuur 7 [GPT-3.5] De gemiddelde testtijd voor verschillende RAG-window groottes en strategieën getest op GPT-3.5-Turbo.



Figuur 8 [Closed-source] Scores voor elk closed-source model.



Figuur 9 [Closed-source] Gemiddelde testtijd per closed-source model.