Do you need the most expensive AI to analyze your documents?

Cost vs. Performance in Document Information Extraction

By Matt Moreno, Stract API

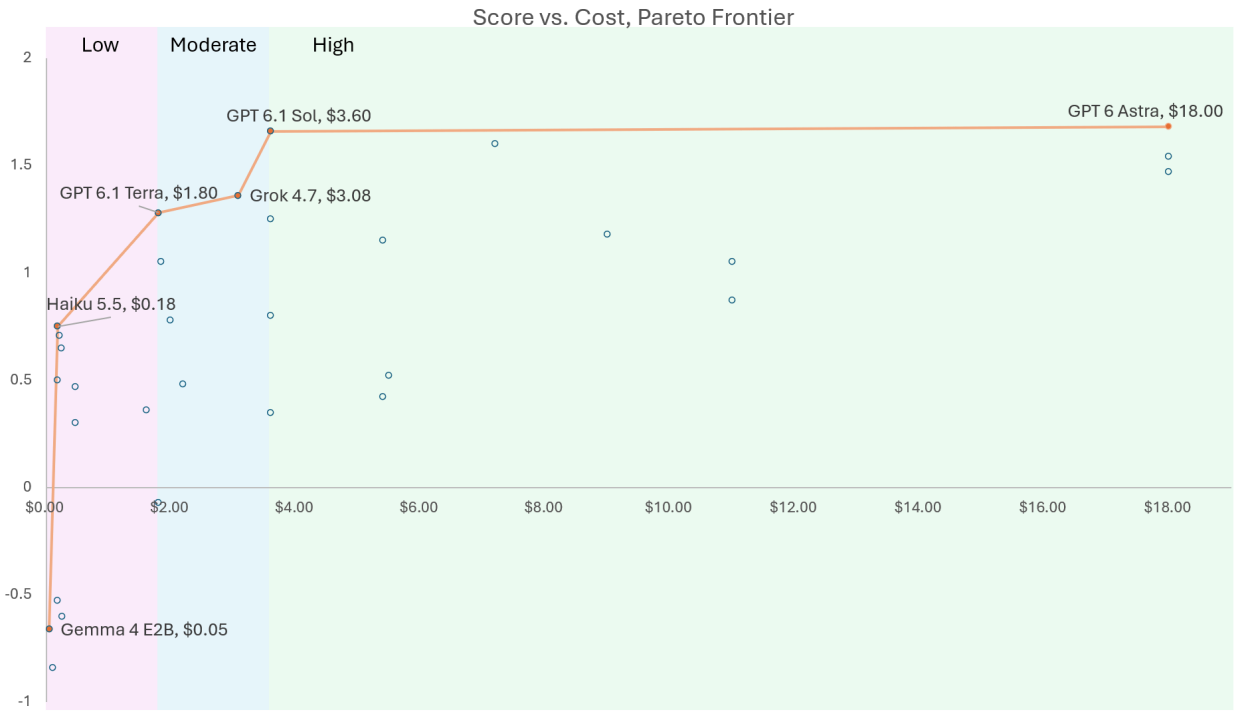
November 2, 2026

(Audience: Stract API users, software engineers incorporating AI into products, operations / finance professionals assessing AI selection)

Summary

We analyzed the performance of many frontier models, both closed and open source, to assess cost-effectiveness for Stract API document information extraction. To do so, we used assumptions relevant to our use-case to simplify how cost is assessed and which models to use. We constructed a performance value based on scores across a number of Artificial Analysis leaderboard tests.

By plotting models based on their cost and performance, we were able to obtain which models are cost optimal in their respective cost buckets.



The most critical insight in this analysis is the Fall 2026 price decreases at both OpenAI and Anthropic for their latest frontier models. We observe that if prices for comparable models (e.g. between GPT 5.6 Terra and GPT 6.1 Terra) had held constant, these models would not have been on the frontier of cost-optimality and would have allowed performant open source models such as GLM-5.3 Flash and Muse Spark Open to be on the cost-optimal frontier instead.

We plan to update this analysis every few months as more models are released.

Jump straight to Section 4 if you prefer to just read the full results.

---

1. The Problem

When we started Stract API, we wanted to simplify the process of using LLMs to solve common business problems. In many ways, we have achieved this goal. However, one question that remained in flux was in some respects the biggest one we needed to answer: how can we make information extraction cost-effective while maintaining performance?

Our Questions

This presents us with three questions:

First: How much does each LLM cost to use?

Next: How do you assess whether a given LLM is good at this specific task?

Finally: Which LLMs should we even be looking at and assessing? And perhaps the related question 3.5: How do you ensure consistent access, especially to the most capable open source models?

Cost

Assessing model cost is challenging for a few reasons. The biggest hurdle is just how varied cost can be.

Just within our dataset, model cost spanned $0.05 / million tokens (/MTok) all the way to $18.00 /MTok, a difference of 360x. Read that again - 360x.

Another challenge is that cost can be broken out into input tokens, output tokens, reasoning tokens, etc. Also, not all models even use the same language to discuss different types out output tokens! This challenge is then compounded by the fact that different models use different numbers of tokens in generating responses. This means that making an apples-to-apples comparison required creating and applying assumptions specific to the use case.

Performance

Assessing model performance is a major challenge as well. Models make headlines by succeeding on a subset of tests, many of which might be good at assessing general capability. But, very few of the most renowned LLM tests are specific or even relevant to Stract's use case. This presented us with the challenge of identifying the tests which are most applicable and relevant to our use case. Then, we had to design a methodology and construct a performance value for each model.

Consistent Access

When creating business workflows, every tool in the pipeline becomes essential to the consistent execution of this process. This has a meaningful impact on how we select which tools to use, as we need tools that we know we can rely on for the medium or even long term. This concept of consistent, reliable, speedy access ends up being a critical guide for how we selected models. First, lets outline the types of models and how they are typically accessed:

- **Closed Source Models:** Can only be accessed via the maker's API. They have historically been more expensive to access, with cost determined by the number of tokens used.

- **Open Source Models:** These have historically been less expensive to access. These can be accessed in a few different ways:
    - By downloading the weights and running the model on a server that you own or rent. Cost is determined by how expensive it is to run the server.
    - Accessed via API from an inference provider or inference marketplace. Cost is usually determined by the number of tokens used.

Because it is not economically feasible to own or rent a server at this time for Stract API, and because managing your own hardware is always risky to begin with, we opted for options which are available via API, which means using only those open source models available through inference providers. This did not impact the selection of closed source models, which are typically only available via API regardless.

Use of Inference Data in Model Training

Due to the sensitivity of data used on the Stract API platform, we required that any inference provider guarantee that the inference data (i.e. the inputs we send and the outputs from the model) will not be used to train other models, as this can become a privacy or security issue in many applications. This is a common constraint in business application. It is notable that many inference marketplaces cannot make this guarantee, thus significantly impacting our analysis options.

A large part of our challenge was sifting through the universe of options and selecting those models which best match our use case.

Our Framing

As we'll discuss more below, it's all too common for teams to just default to what the public considers the "best" models (often correlating to the "most expensive"), hoping that replacing expensive humans with "cheaper" (but still comparatively pricey) machines will immediately result in efficiency gains.

While this is sometimes the case, it's more common that teams are stuck with outrageous bills with very few efficiency gains to speak of. In some cases, as we're hearing on the news, companies are spending even more on development with lower-quality or less output. As a company that purports to be an expert in the topic, we wanted to move away from these popular assumptions about what is "best," and ground our professional recommendations in data based on a closer examination of the concrete facts, looking at each model and its performance on equal footing without relying on reputation and marketing.

---

2. The Stract Use Case

At a high level, the first thing that Stract API does is to assess a document text along with what pieces of information that users want to extract from the document. For example, a procurement team may ask "Does this contract auto-renew?" for each contract they handle. Stract then runs the documents and desired information through LLMs, and ensures that valid, structured answers are emitted from the LLM as output. That output is then collated across all the uploaded documents and turned into usable CSV / JSON. For non-technical users those CSV / JSON files can be directly opened in Excel / Google sheets, or copied into a table in Word / Google Docs, or even uploaded directly to that user's primary LLM for analysis.

This implies the following for our most critical analysis variables in choosing what models to recommend and make available:

Model Choice

- Must allow users to specify an AI model of their choice from a large list of capable models.
- Models must be available on-demand. Any given model will not necessarily always be in use, meaning hosting our own models is not economically feasible, while access must be guaranteed.
- Latency (time to result) must be low to fit into common business workflows.
- Inference data must not be retained for training or sharing with 3rd parties

Model Inputs Include

- Document text
- Output data schema (i.e. definition of what information is being requested)
- General instructions

Given the above, model inputs are constant, and do not need to be addressed as part of this analysis.

Model Outputs Include

- Structured data that conforms to schema
- Supporting evidence for each data point
- Other output is intentionally limited to reduce cost
- In testing models for our use case, including reasoning tokens in the number of output tokens does not change the ratio of input to output tokens meaningfully

Given the above, model outputs are almost constant across models, and will not be addressed as part of this analysis.

3. Methodology

3.1 Model Selection

- Given the economic infeasibility of hosting, we narrowed our list of models to the list of models available on-demand from large inference service providers (AWS Bedrock, FireworksAI, Anthropic, OpenAI, etc.). Large inference service providers in general also provide low latency on requests, making use in most business workflows possible.
- Many older models did not have test values (see below 3.3 Model Performance for description of tests), thus have been excluded from the analysis for lack of data.
- Selected models include:
    - (Anthropic) Claude Fable 5
    - (Anthropic) Claude Haiku 4.5
    - (Anthropic) Claude Opus 4.8
    - (Anthropic) Claude Sonnet 4.6
    - (Anthropic) Claude Sonnet 5
    - (Anthropic) Fable 5.1
    - (Anthropic) Haiku 5.5
    - (Anthropic) Opus 5.5
    - (Anthropic) Sonnet 5.5
    - (DeepSeek) DeepSeek V4.1 Flash 1017
    - (Google) Gemma 4 31B
    - (Google) Gemma 4 E2B
    - (Meta) Muse Spark Open
    - (Minimax) Minimax M3
    - (Minimax) Minimax M3.1 Flash
    - (Moonshot AI) Kimi K3
    - (Nvidia) Nemotron Nano 3 30B
    - (Nvidia) Nemotron Super 3 120B
    - (OpenAI) GPT 6 Astra
    - (OpenAI) GPT 6 Luna
    - (OpenAI) GPT 6.1 Sol
    - (OpenAI) GPT 6.1 Terra
    - (OpenAI) GPT-5.5
    - (OpenAI) GPT-5.6 Luna
    - (OpenAI) GPT-5.6 Sol
    - (OpenAI) GPT-5.6 Terra
    - (Qwen) Qwen 3.8 Max
    - (Thinking Machines) Inkling

- (xAI) Grok 4.7
- (Z AI) GLM-5.3
- (Z AI) GLM-5.3 Flash

3.2 Cost Assumptions

- In testing Stract API, we have seen a ratio of about 80% inputs tokens to 20% output tokens (including reasoning tokens) by count.
- This ratio is strongly dependent on how Stract API is being used, however, and can reach as low as 60/40 for very short documents or 95/5 for very long documents.
- To give us one number to work with, we will select 80/20 and combine input and output token costs into one weighted cost value.
- The majority of models have an output-to-input cost ratio of 5, though some go as low as 2 and others as high as 8. In general, you can expect a ratio between 4-6, meaning that the choice of weights used for input to output token count doesn't make a huge difference, and in general doesn't seem to greatly change the results of this analysis.

3.3 Model Performance

- We utilized Artificial Analysis, a large collection of independently run leaderboards for a wide range of LLM capability tests. Specifically, we use the scores from four of the tests available on this platform, one of which provides two metrics:

Benchmark	**What it adds to the score**	**Weight**
GDP.pdf[1]	Performing tasks on business-relevant documents	35%
AA-Omniscience v1.1: Accuracy[2]	Answering questions accurately	20%
AA-Omniscience v1.1: Hallucination	Not hallucinating an answer	10%
AA-LCR v1.1: Long Context Reasoning[3]	Answering questions about business-relevant	20%

[1]"GDP.pdf is a benchmark from Surge AI for realistic professional work over long PDF documents. Its 100 tasks span ten domains, 4,592 source pages, and 1,275 atomic grading criteria." (Artificial Analysis)
[2]"AA-Omniscience is a knowledge and hallucination benchmark that rewards accuracy, punishes bad guesses, and provides a comprehensive view of which models produce factually reliable outputs across different domains." (Artificial Analysis)
[3]"LCR features 100 questions across diverse document types including academic papers, company financials, government consultations, legal documents, industry reports, and marketing materials, requiring genuine reasoning rather than simple data extraction." (Artificial Analysis)

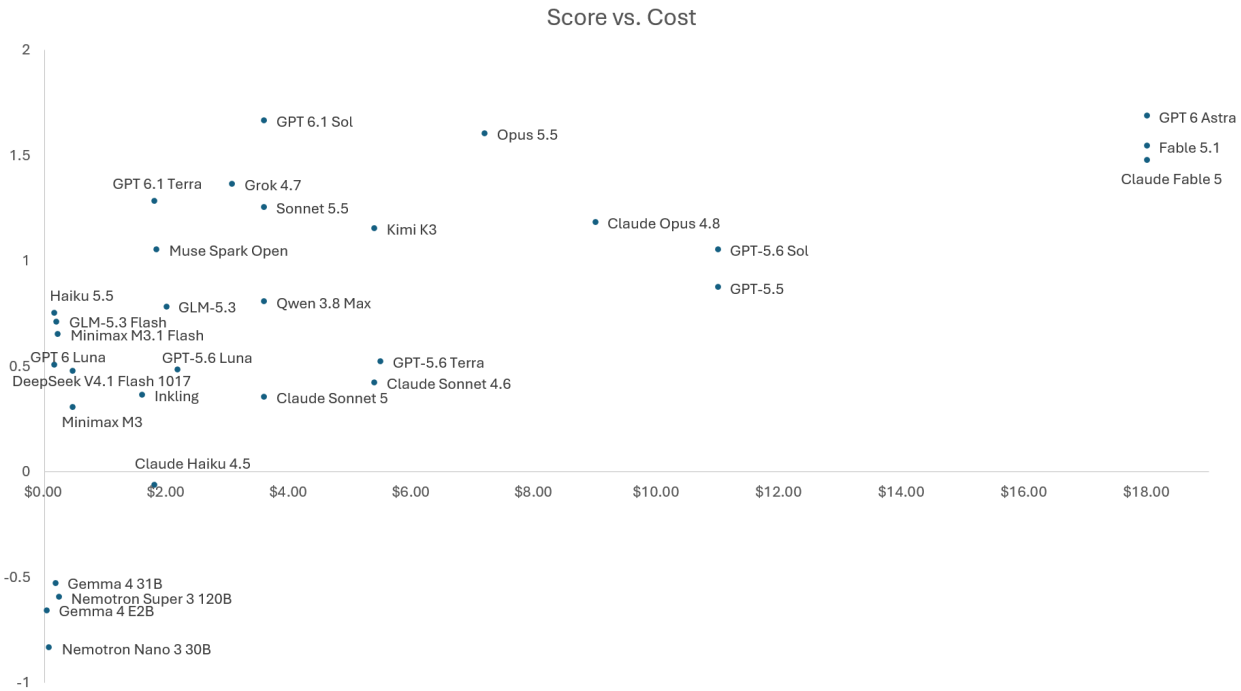
Benchmark	What it adds to the score	Weight
	documents, especially very long documents	
AA-Briefcase v1.1: Agentic Knowledge Work[4]	Performing business-relevant tasks involving documents	15%

- We performed a Z-score normalization on the scores (based on the entire universe of models) in order to obtain comparable values across tests. These Z-score normalized values were then combined using the aforementioned weights to obtain the final model performance score.

---

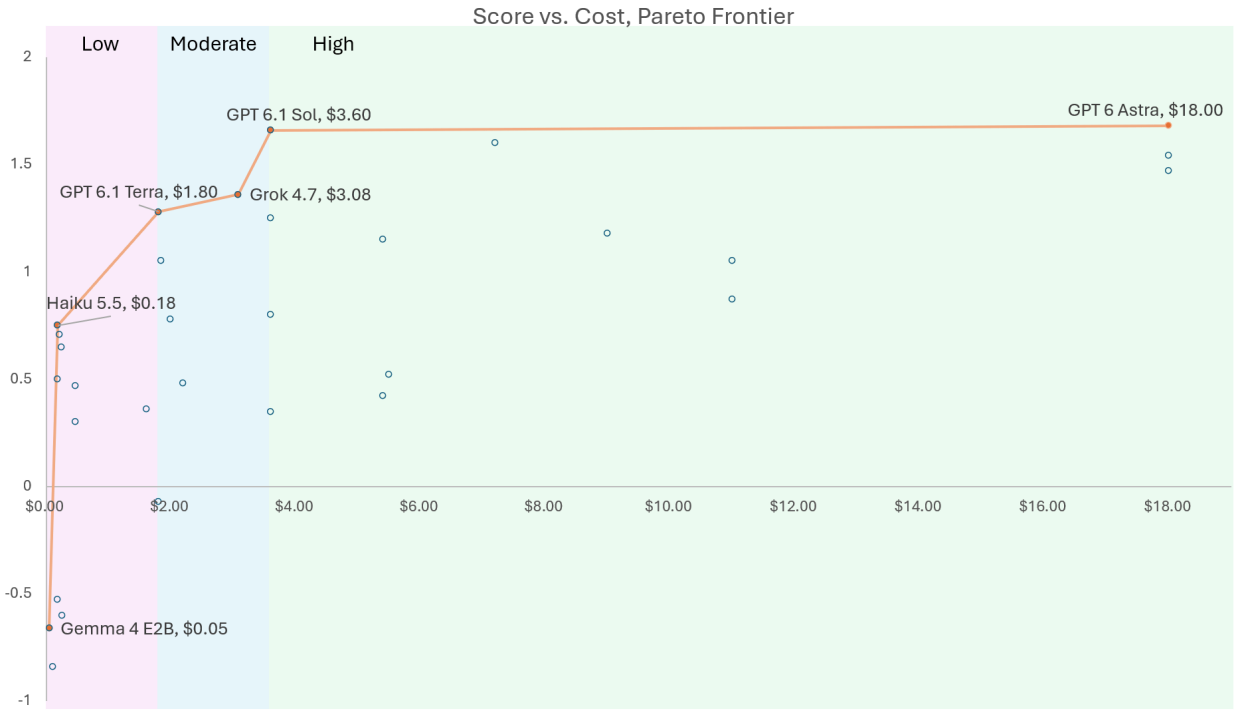
4. Results & Analysis

Full Results



[4]"AA-Briefcase evaluates models across four multi-week knowledge work projects, comprising thousands of input files and 91 tasks in total. Across the scenarios, models must complete realistic professional workflows in fields such as data science, product management, and corporate strategy." (Artificial Analysis)

Pareto Frontier (Cost Optimal Choices) and Cost Buckets



4.1 Choosing Cost Buckets

- We utilize the Pareto frontier (i.e. the cost optimal choice) to help guide the determination of cost buckets. A model is on the frontier if no other model in the analysis offers a higher score at the same or lower cost. Any model below the frontier is matched or beaten on both cost and score by a model on it.
- Six models are on the frontier: Gemma 4 E2B, Haiku 5.5, GPT 6.1 Terra, Grok 4.7, GPT 6.1 Sol, and GPT 6 Astra.
- We set our bucket cutoffs at $1.80 /MTok and $3.60 /MTok, the prices of GPT 6.1 Terra and GPT 6.1 Sol. Each of these models offers a substantial score increase over the previous bucket's recommended model: Haiku 5.5 scores 0.75, GPT 6.1 Terra scores 1.28, and GPT 6.1 Sol scores 1.66.
- We do not add a bucket above $3.60 /MTok because the frontier flattens out there. GPT 6 Astra, the only frontier model above that price, costs five times as much as GPT 6.1 Sol for a score increase of just 0.02.

4.2 Cost < $1.80 /MTok

- In the under $1.80 per million tokens (/MTok) category, Anthropic's Haiku 5.5 was the optimal choice, coming in at just $0.18 /MTok and performing at a weighted average of 0.75 StDev above the mean.

- We do see Gemma 4 E2B on the frontier of performance at a very, very low $0.05 /MTok, but with such low performance (0.66 StDev *below* the mean), we cannot recommend using this model for anything but the simplest tasks.

4.3 $1.80 /MTok <= Cost < $3.60 /MTok

- In the < $3.60 category, we have two strong contenders with OpenAI's GPT 6.1 Terra and xAI's Grok 4.7. Here, we would lean towards GPT 6.1 Terra, which comes at a much cheaper price point than Grok 4.7 (40% cheaper) with similar performance (1.28 vs. 1.36).

4.4 $3.60 /MTok <= Cost

- There are a number of high performing models in this group, but we would consider OpenAI's GPT 6.1 Sol to be the clear winner when weighing cost and performance. GPT 6.1 Sol comes in at the lowest price point in the group with $3.60 /MTok and has the second-highest performance at 1.66 StDev above the mean. The only model with higher performance is GPT 6 Astra at 1.68, but at a whopping $18.00 /MTok, we cannot recommend this model unless you absolutely need the additional capability.
- It is also notable that while GPT 6 Astra is known as an absolutely superb model in general capabilities, and is known to be much more capable than GPT 6.1 Sol, for the index constructed here, it does not necessarily shine above the rest. This seems to be equally true of other frontier models, whose score in this analysis does not seem to set them far above other cheaper models.

4.5 Insensitive to Input vs. Output Ratio

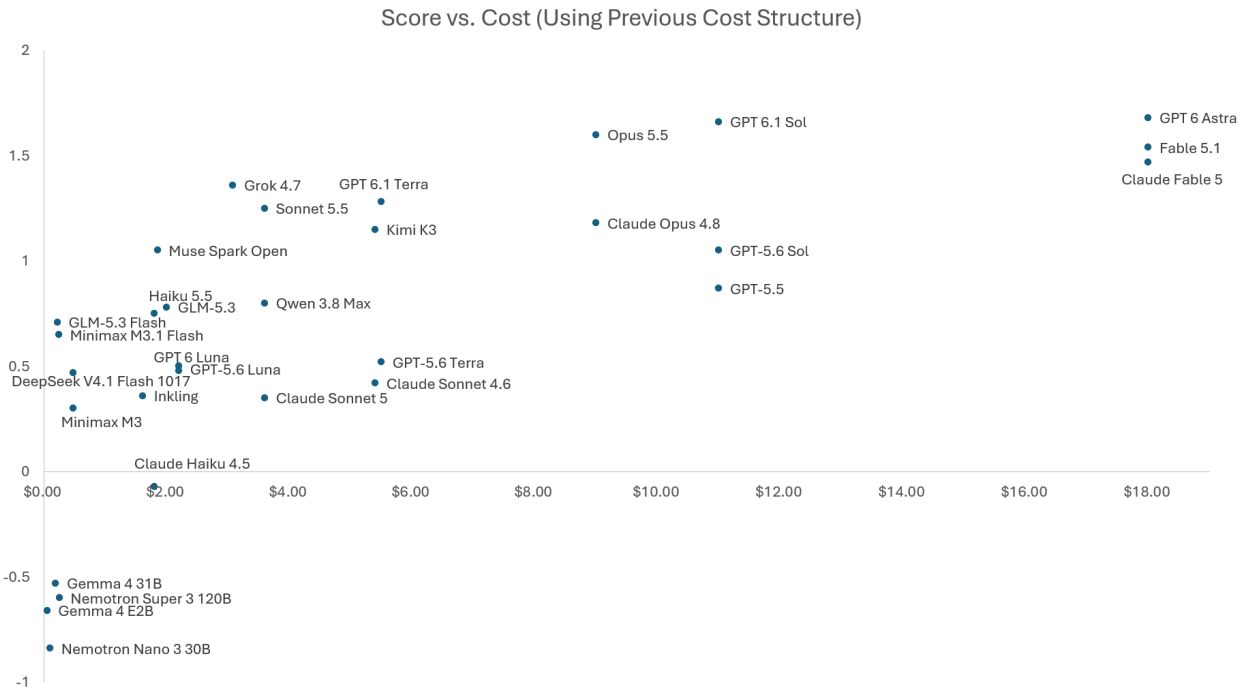
- This analysis is relatively insensitive to the ratio of input to output tokens, which as mentioned before was assumed to be 80/20. If we change that to the extremes of 95/5 and 60/40, our findings remain exactly the same with one exception: Grok 4.7 is no longer on the frontier at 95/5 input to output token ratio. In other words, because of Grok's price structure, having more input tokens makes it less cost-efficient relative to its score in this analysis.

4.6 Fall 2026 Price Decreases

- One extremely notable point here is how pricing strategy has seemingly changed at Anthropic and OpenAI recently, making their products much more competitive. Compare the cost of GPT 5.6 Sol to GPT 6.1 Sol ($11.00 vs. $3.60), GPT 5.6 Terra to GPT 6.1 Terra ($5.50 vs. $1.80), or Haiku 4.5 to Haiku 5.5 ($1.80 vs. $0.18).
- Without these price changes, some of these models would not have been at the cost / performance frontier in this analysis (see graph below). We believe these price changes occurred within the last few months (September - October 2026), so we should keep an

eye on prices across the board, as there might be an industry-wide move towards more competitive pricing.

- This would have allowed performant open source models to be on the frontier, such as GLM-5.3 Flash and Muse Spark Open. It also would have allowed Grok 4.7 to be a clear candidate for cost-optimality, rather than the current also-ran.
- It's notable that GPT 6.1 Sol's strong performance would have kept it on the cost-optimality curve, though at a much higher price point.



---

5. Practical Framework for Choosing a Model

- **Classify the task:** For Stract API users, we will aim to keep this simple. Based on our recommendations above, we will be providing a simple "Low Complexity", "Moderate Complexity", and "High Complexity" choice, representing a user's assessment of how complex the questions they are posing to the model are. This will select the cost appropriate model: Haiku 5.5 for low complexity, GPT 6.1 Terra for moderate complexity, and GPT 6.1 Sol for high complexity.
- **Start cheap, measure up:** Users should start with the "Low" complexity choice first, and if they are not getting accurate results, they should switch to "Moderate" or "High" to obtain better results.
- For users with a more specific preference, we will still allow you to select any model of your choice from the list above.

- **Re-evaluate periodically:** Prices drop and models improve. Today's frontier is next year's mid-tier. We will of course keep updating these results to ensure we provide the highest efficiency possible.

---

6. Limitations and Future Work

- Point-in-time snapshot: this paper assesses model performance and prices as of *November 2, 2026*.
- Budget is a big limiting factor; with a larger budget, we could run a more specific analysis and develop our own performance metric. However, given budget constraints we feel confident in the values used from Artificial Analysis being a proxy for the general use case of Stract API.
- Though it would add significant complexity to this analysis, it might be helpful to do a deeper dive into the assumptions around cost, such as the ratio of input cost vs. output cost, the ratio of input tokens vs. output tokens, etc. These are some of the largest assumptions made in this analysis and are central to the findings.

Disclosure: Stract API is a customer of the service providers mentioned in this post, including AWS, FireworksAI, Anthropic, and OpenAI, but has no other financial relationship with them, nor was given any financial incentives for the writing of this paper. Stract API has no financial ties to Artificial Analysis.

---

7. Get Started with Stract API

Stract API makes it simple to get information out of your documents, without needing to become an expert in LLMs along the way. We've done the homework on which models deliver, so you get the highest performance at a competitive cost and can stay focused on your work. Get started with Stract API today.

---

8 Full Results Table

Model	**Cost (/MTok)**	**Score**
Claude Haiku 4.5	$1.80	-0.07
Claude Sonnet 5	$3.60	0.35

Model	Cost (/MTok)	Score
Claude Sonnet 4.6	$5.40	0.42
Claude Opus 4.8	$9.00	1.18
Claude Fable 5	$18.00	1.47
Gemma 4 E2B	$0.05	-0.66
Gemma 4 31B	$0.19	-0.53
GPT-5.6 Luna	$2.20	0.48
GPT-5.6 Terra	$5.50	0.52
GPT-5.6 Sol	$11.00	1.05
GPT-5.5	$11.00	0.87
Kimi K3	$5.40	1.15
Nemotron Nano 3 30B	$0.10	-0.84
Nemotron Super 3 120B	$0.25	-0.60
Grok 4.7	$3.08	1.36
DeepSeek V4.1 Flash 1017	$0.48	0.47
Qwen 3.8 Max	$3.60	0.80
GLM-5.3	$2.00	0.78
GLM-5.3 Flash	$0.22	0.71
Minimax M3	$0.48	0.30
Minimax M3.1 Flash	$0.24	0.65
Muse Spark Open	$1.85	1.05
Inkling	$1.61	0.36
Haiku 5.5	$0.18	0.75
Sonnet 5.5	$3.60	1.25

Model	Cost (/MTok)	Score
Opus 5.5	$7.20	1.60
Fable 5.1	$18.00	1.54
GPT 6 Luna	$0.18	0.50
GPT 6.1 Terra	$1.80	1.28
GPT 6.1 Sol	$3.60	1.66
GPT 6 Astra	$18.00	1.68