Blog

LLM Benchmarking: How to Find the Ideal Large Language Model for Your Needs

13 min read

Choosing a large language model is no easy task. There are ways, however, to objectively measure the performance of the various available options: benchmarking is the magic word when it comes to finding the right model in a dynamic market.

management, these LLMs hold enormous potential to boost the flow, utilization, and provision of data, both internally and in customer interactions.

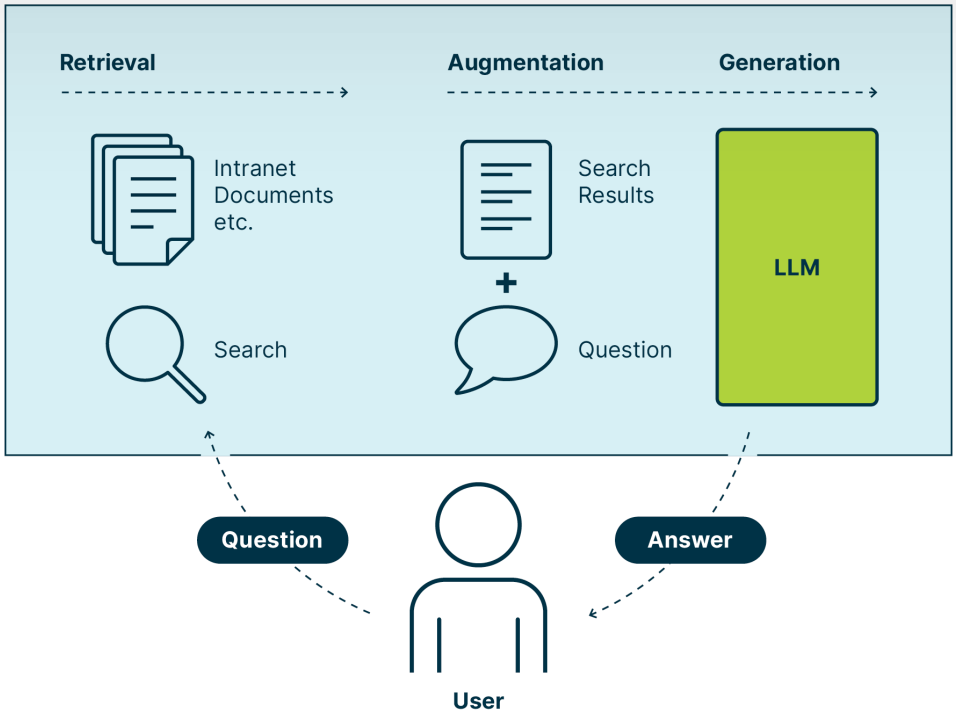
But this technology comes with challenges that cannot be ignored: On the one hand, there are still concerns about the protection of data entrusted to an LLM. On the other, all known LLMs focus on English, to the detriment of other languages. And finally, prospective users have to choose from a broad range of models that differ considerably regarding their general performance and individual strengths.

For organizations wanting to explore the business case of LLMs, a possible approach is to develop a chatbot for internal use among employees as a first step to conduct research, gather experience with this technology, and test various models. With this goal in mind, we want to create a chatbot that, when prompted by a user query, combs countless documents and texts in the company intranet for relevant information and generates an accurate and concise response presented in natural language. The technological foundation on which the chatbot shall be based is retrieval-augmented generation (RAG).

What is retrieval-augmented generation?

RAG is a process that extends the capabilities of LLMs with an information retrieval system. The LLM relies on this external knowledge base when responding to a query. This ensures full control over where the LLM obtains the information for its answers.

Using our chatbot as an example, we can demonstrate the two phases of the RAG process:



1. In the first step («retrieval»), the chatbot searches the intranet and collects the documents and information relevant to the search query.
2. Then the initial question is supplemented with these search results («augmented») and handed over to the LLM, which processes the provided data and generates an answer.

The existing search function of our intranet is perfectly sufficient for the first stage of this process. For the second, however, we will need an LLM.

What is a Large Language Model?

LLMs are AI models that specialize in tasks related to human language. They are trained by means of enormous bodies of text and are thus able to understand and generate natural language.

GPT-4o	Claude 3 Opus	Mistral Large	Gemini 1.5 Pro	Llama 3 70B Instruct
OpenAI	ANTHROP\C	MISTRAL AI_	Google	Meta

Now we are faced with the task of finding out which LLM is best suited to our RAG use case. The available options are the models of the leading providers OpenAI, Anthropic, Mistral, Google, and Meta:

- OpenAI: GPT-4o
- Anthropic: Claude 3 Opus
- Mistral: Mistral Large
- Google: Gemini 1.5 Pro
- Meta: Llama 3 70B Instruct

To find the best model for our purposes, we need to develop a precise evaluation methodology. An important aspect here are the non-functional and functional criteria.

Non-functional criteria

Functional criteria

At a fundamental level, we can differentiate whether the model is open source or commercial. Of the models listed above, only Llama 3 70B Instruct (meta license) is open source, the rest are commercial offerings. This has an impact on the non-functional criteria, in particular the provision and operation of an LLM. Here, there are two options:

- API only: The LLM is made available by the provider and is only accessible to us via API.
- Own deployment: We provide the LLM ourselves and retain full control over it.

All models offer provision of the LLM via API, but own deployment is only possible with open-source models. This means that we must accept certain compromises when going the commercial route. For example, if we operate and control the LLM ourselves, we can rest slightly more reassured about the security of our data than with an external LLM. In this case it is particularly important that we carefully study the data processing agreements of the respective provider to avoid any unpleasant surprises.

In addition, all commercial models have a limited life cycle. Transitions from one model to the next are particularly evident with OpenAI. As soon as such a life cycle is completed, we as users have no choice but to move on, adopt the successor model, and deal with the impact on our operations. In this context, it is important to understand that an LLM differs from other parts of a software system at a conceptual level. Databases or application servers can be replaced so seamlessly that users are left none the wiser. But if an LLM is replaced, even if only with a newer version, users generally take notice: the system will most probably provide a different answer to the same question, which may have the same core statement but will not be worded identically.

We will touch upon these non-functional criteria, i.e., control, data protection, and stability, as well as the important question of costs, once again later on. But first

Functional criteria

Our vision of an LLM is easily explained: We want the LLM that shows the best results when answering our questions. However, a surprising amount of effort is required to perform this evaluation, as we are dealing with a number of unknowns here. We are unaware of the users' questions, and therefore don't know the results of the retrieval phase, nor have we defined the right answers to these questions.

We therefore need a more general criterion, which we define as follows: Which LLM shows the best results in answering several questions about a specific context, e.g., a Wikipedia article? Both the questions and the context must be in German (which will pose an additional challenge).

So, to determine the best model, we need so-called benchmarks.

What is a benchmark?

In the broadest sense, a benchmark is a means to measure the performance of products, processes, or organizational structures in relation to a reference value. Benchmarking is routinely applied in computer and information technology, where hardware components, such as processors and graphics cards, and software applications are put to the test with benchmarks.

Benchmarks have also played a key role since the earliest days of machine learning (ML). Only they make it possible to compare the quality of different ML models and measure their progress.

Existing benchmarks

To create a benchmark, you need a) a data set, b) a metric, and c) the models that the benchmark is to test.

Let's take a look at the MNIST dataset as an example from the early days of ML. It consists of pictures of handwritten numbers (a) and the corresponding numerical values (also known as "ground truth"). A few samples from the MNIST test dataset (image by Josef Steppan, CC BY-SA 4.0):

In the MNIST benchmark, two image recognition models (c) are supposed to identify the handwritten numbers in the images. The applied metric (b) is simple: If the number given by the model matches the ground truth, the identification has been successful. The higher the number of correct identifications, the better the model.

There are also numerous benchmarks for LLMs that focus on the various use cases of LLMs, e.g., translating texts from English into German, summarizing longer documents, or solving mathematical problems.

To find the best large language model for us, we will focus on the benchmarks where the dataset corresponds to our RAG use case. In other words, it consists of contexts, questions, and answers (ground truth). The LLM is supposed to solve

This type of task is referred to as open-book question answering, as the model is provided with the information required to answer the question. In contrast, closed-book question answering states that the model must answer the question without additional information.

Here are some well-known benchmarks for open-book question answering:

- NarrativeQA is an English-language dataset of stories and related questions designed to test reading comprehension, especially of long documents.
- HotpotQA is a dataset with Wikipedia-based question-answer pairs. To answer the questions, the LLM must find several paragraphs and understand their content.
- DROP is a reading comprehension benchmark that requires reasoning over several paragraphs. The answers usually consist of a number or a name, allowing a simple metric.

LLMs and technical reports

Information on a new LLM is usually published as part of a technical report. It details the performance of the model in various benchmarks and, in some cases, comparisons with the results of other models.

Even with an English-language RAG system, the problem with these reports is that they do not contain the results of the LLMs for a commonly applied question-answering benchmark. While the reports of Anthropic's model Claude 3 and GPT-4o of OpenAI both contain an F1-score for the DROP benchmark, it is absent from the documentation for Mistral's Large model.

Our own benchmark

A quick inspection of platforms that provide datasets (e.g., Hugging Face, Kaggle, and Papers With Code) dashes any hope we may have had of finding an easy solution. Only very few German-language datasets for our open-book question answering are on offer here, and even the most promising candidate leaves a lot to be desired. The individual answers, i.e., the ground truths, vary widely from each other: sometimes they are complete sentences, sometimes sentence fragments, sometimes even just single words. This inconsistency among the responses makes it extremely difficult to define a meaningful metric that can help compare the results of the individual LLMs.

Accordingly, we have to look elsewhere for a suitable dataset. One option is to create such a dataset ourselves – but this requires considerable manual effort and costs a lot. A more viable alternative is to machine-translate an existing English-language dataset into German. The DROP data set stands out as a convenient choice. Its original task corresponds roughly to a RAG system, and it has an established dataset, which means it is used for many models.

The DROP dataset

The DROP (Discrete Reasoning over the Content of Paragraphs) dataset is created manually (i.e., by crowdsourcing) and comprises around 96,000 questions. Answering the questions requires understanding several paragraphs of a Wikipedia article. The dataset contains numerous questions for each Wikipedia article, the latter being the input context for the LLMs.

A machine-translated example:

Context: *Auf das gesamte County bezogen setzte sich die Bevölkerung zusammen aus 24,90% Einwohnern unter 18 Jahren, 7,20% zwischen 18 und 24 Jahren, 28,10% zwischen 25 und 44 Jahren, 23,60% zwischen 45 und 64 Jahren und 16,30% waren 65 Jahre alt oder darüber. Das Durchschnittsalter betrug 39*

(Original: In the county, the population was spread out with 24.90% under the age of 18, 7.20% from 18 to 24, 28.10% from 25 to 44, 23.60% from 45 to 64, and 16.30% who were 65 years of age or older. The median age was 39 years. For every 100 females there were 97.10 males. For every 100 females age 18 and over, there were 93.60 males.)

Question: *Wie viele Prozent der Bezirksbevölkerung waren nicht zwischen 18 und 24 Jahre alt*

(Original: How many percent of the county population were not from 18 to 24?)

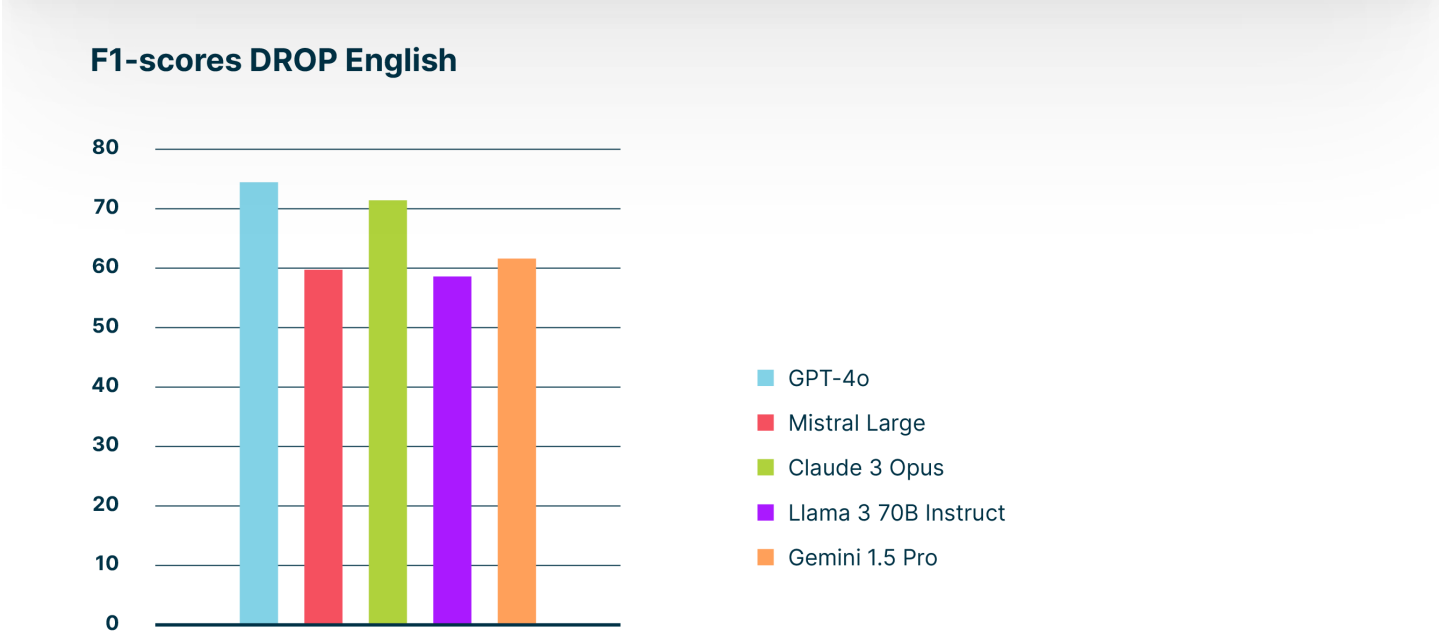
Answer: *92,8%.*

For our German DROP benchmark, we will select 1000 context-question-answer units and translate them into German. The selection criteria are a) no questions about sports and b) no multiple-choice answers. The context and the question-answer pairs will be translated by the translation service of AWS.

Apart from the dataset, we also need a metric, i.e., a standardized procedure with which we can compare the answers of the LLMs with the ground truth. For this purpose, we will use the usual metric for the DROP benchmark, the F1-score. This is the mean value of two variables: precision on the one hand and recall on the other. Precision indicates how many words in the LLM's answer are included in the ground truth, recall states how many words of the ground truth are contained in the LLM's answer. The F1-score is a suitable metric for the DROP dataset, as the correct ground truth answer is only made up of numbers, names, or very simple sentences.

Results in the English LLM benchmark

To run an initial test of our benchmarking concept, we will create a benchmark with the 1000 selected data points in their original English-language version. For

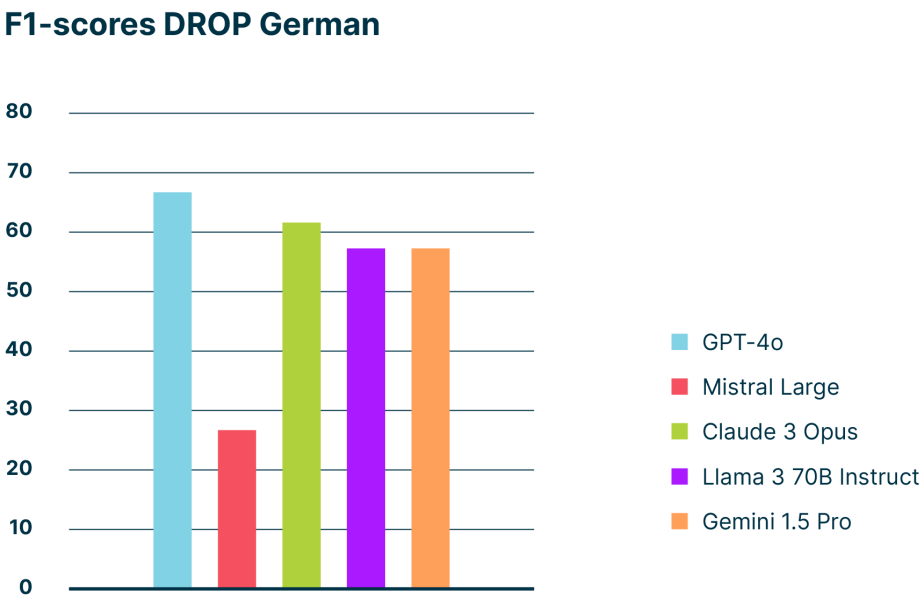


The difference between the F1-scores can be explained by the fact that three examples of context-question-answer units were given in the prompt templates (3-shot), which isn't the case with our templates. The F1-score in the technical report of Google's Gemini model is not suitable for a comparison, as it is not entirely clear how it comes about.

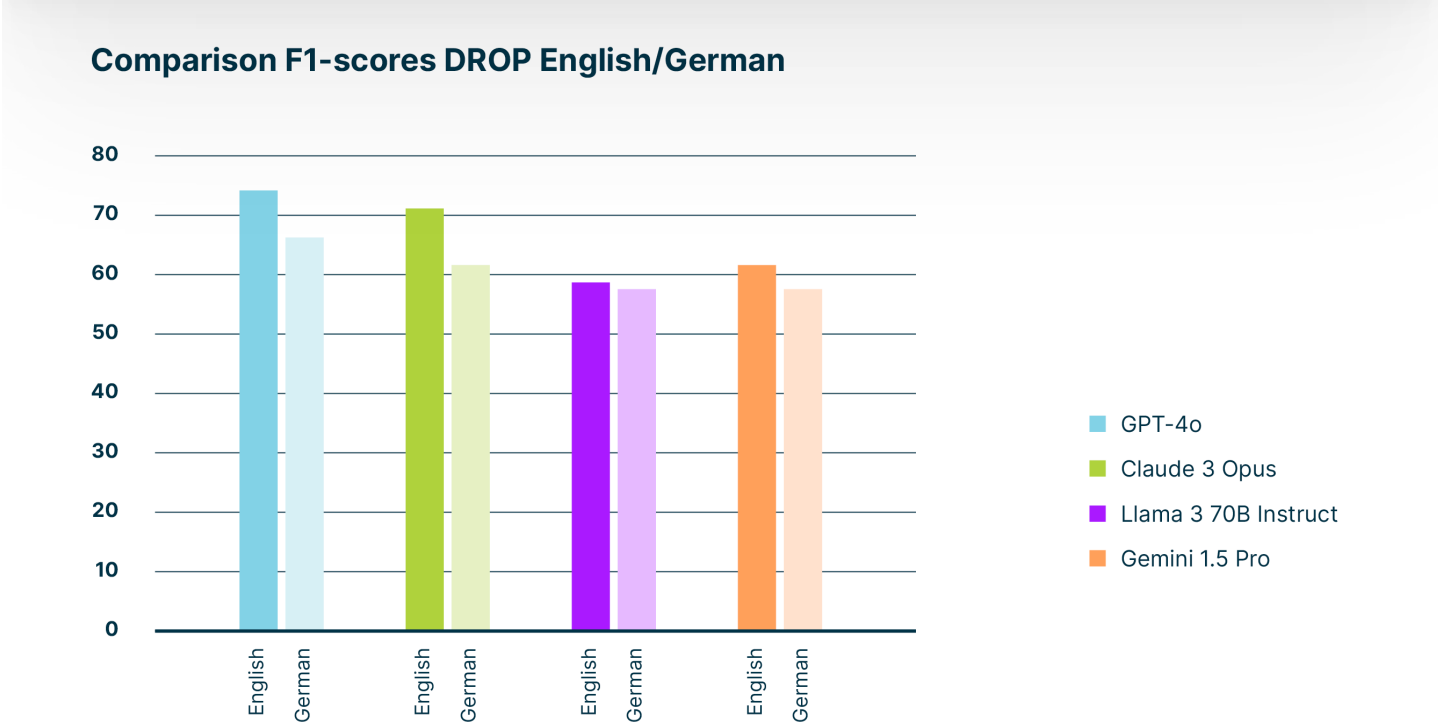
F1-score	GPT-4o	Claude 3 Opus	Mistral Large	Gemini 1.5 Pro	Llama 3 70B Instruct
Technical report	83.4	83.1	–	78.9	79.7
Our benchmark	74.4	71.2	59.3	61.3	59.3

The best models are, in descending order, GPT-4o, Claude 3 Opus, and Gemini 1.5 Pro.

Let us now turn to our actual objective, benchmarking with German-language data. The setup for determining the F1-score remains the same, except that all English-language data and prompt templates are machine-translated into German. The result:



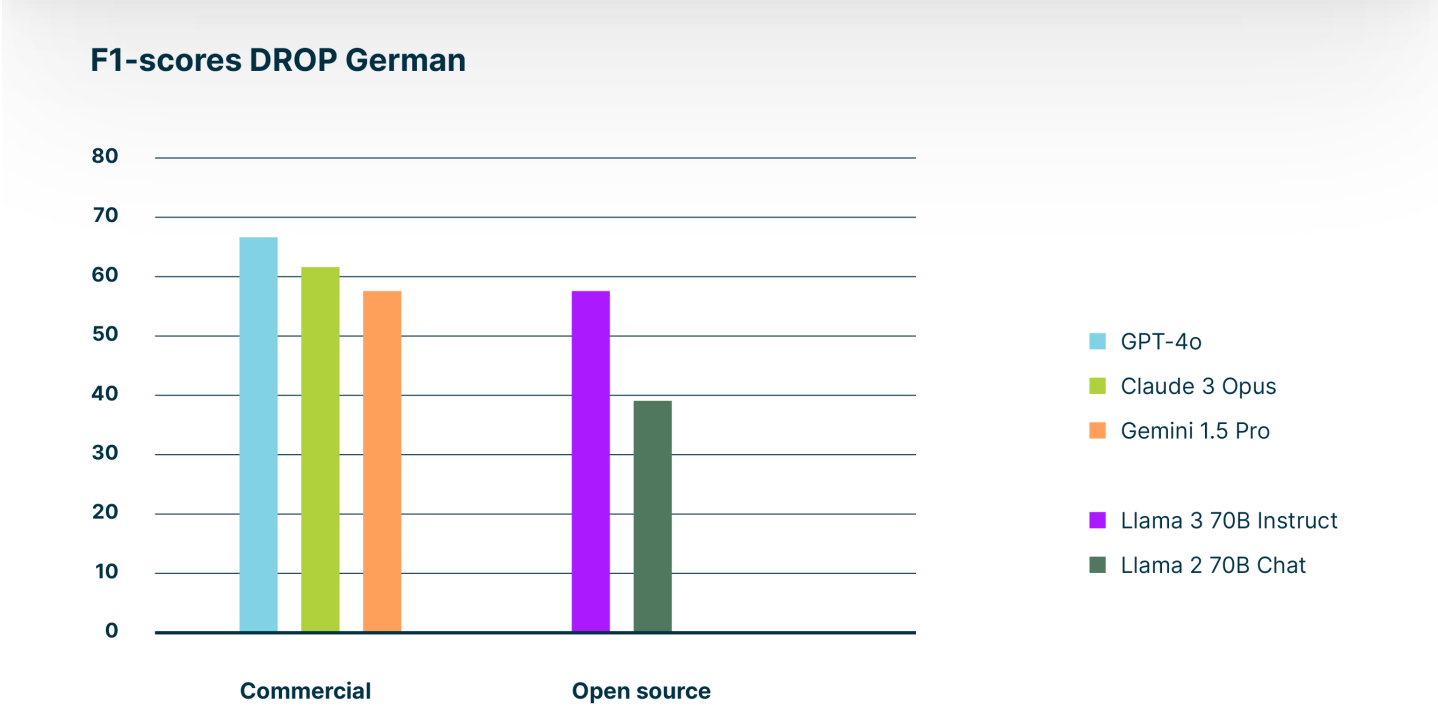
GPT-4o and Claude 3 Opus once again claim the first and second place, respectively. The third place is now shared by Gemini 1.5 Pro and the open-source model Llama 3 70B Instruct. As was to be expected, all models achieved worse scores in the German benchmark than in the previous English benchmark, which is due to the predominantly English training data:



We recorded the smallest difference between the English and German test results with the Lllama 3 model, which also stands out for another reason.

High dynamics

Llama 3 70B Instruct is the most recent of the LLMs we tested. In the first runs of our benchmarking, we still used the predecessor model Llama 2 70B Chat, allowing a comparison with its successor:



This jump in performance is astonishing and has fundamentally changed the situation: Previously, only commercial models were represented in the top group of our LLM ranking. Now a competitive open-source LLM is available with Llama 3 70B Instruct.

From the user's point of view, the intense competition among the various providers is very positive – provided you have an objective means of comparing the different offerings, such as our benchmark.

Non-functional criteria

Now that we have determined the performance of the models in our use case example, we need to consider the non-functional criteria mentioned above, which also play an important role in our decision-making process. Here is a brief overview of the basic conditions of the various providers and models:

Model	Deployment	Data

OpenAI	GPT-4o	**Azure** (Switzerland North) **OpenAI**	Enterprise-ready	0.007	Low
Mistral	Mistral Large	**AWS EU** (Paris) **Azure EU** (Paris) **La Plateforme** (Mistral)	Enterprise-ready	0.0146	Low
Meta	Llama 3 70B Instruct	**AWS** US-East **Own deployment**	Enterprise-ready	0.0021	High
Anthropic	Claude 3 Opus	**AWS** US-West **Claude.ai** platform **Google Cloud** (announced)	Enterprise-ready	0.027	Low
Google	Gemini 1.5 Pro	**Google Cloud**	Enterprise-ready	0.0049	Low

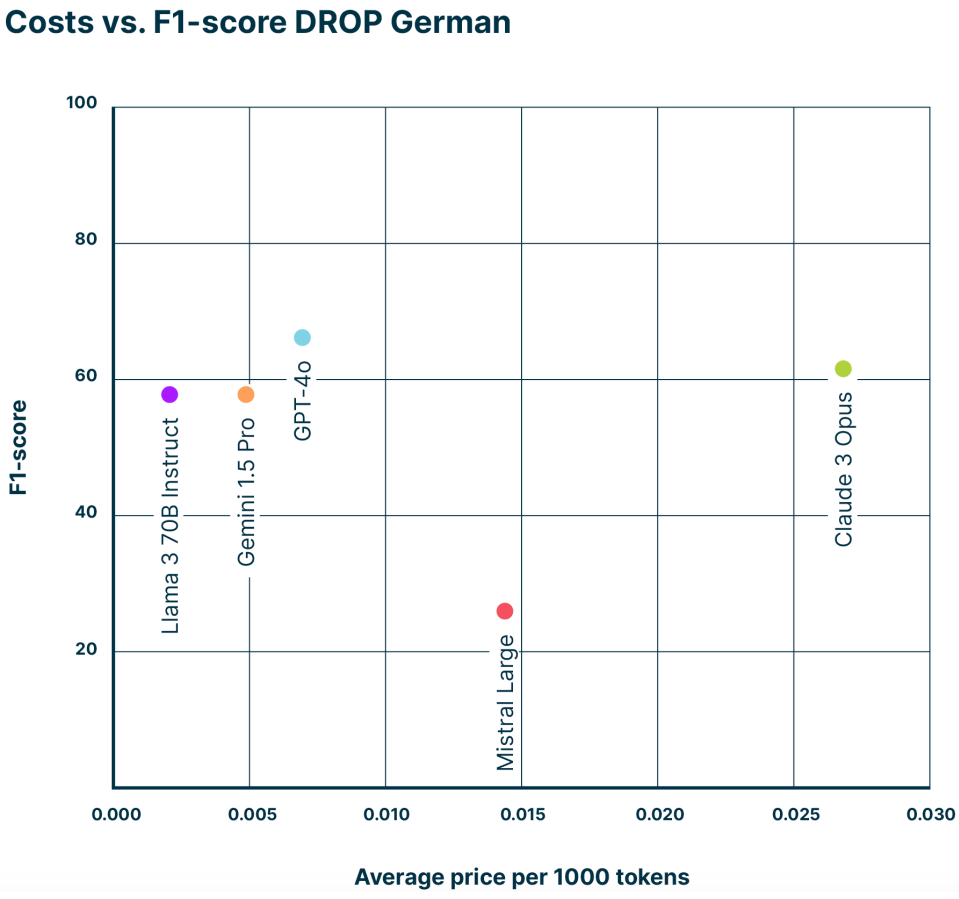
For the deployment options, we initially distinguished between API only and own deployment. In the deployment options column, Azure or AWS refers to the providers for the API only option. The selected provider determines the other

The costs are calculated per number of tokens for all models. A token can be thought of as a partial word. Both the tokens of the prompt (i.e., of the input) and the tokens of the generated response (output) are subject to a charge and differ in price. To enable a cost comparison, we will calculate the average price per token with the assumption that the prompt contains four times as many tokens as the response. The table above shows the prices in US dollars per 1000 tokens.

The data processing agreements with the providers Google Vertex AI, Amazon Bedrock, and Azure are essentially comparable. The data protection requirements for the use of an LLM are met by these providers.

Cost-performance comparison

If we add the costs as a further quantitative criterion to the results achieved in the benchmark (F1-score), the following picture emerges:



Selecting a model

To finally select an LLM, we need to weigh up the functional and non-functional criteria. We'll illustrate how this process can work with two scenarios for our use case, a RAG-based chatbot:

Scenario A: Employees use the chatbot to answer customer queries. The generated responses were tested extensively before launch. Fully automated responses to customer queries are planned for a subsequent expansion phase.

Scenario B: The chatbot serves employees as an additional information retrieval option alongside the intranet search.

In scenario A, the lack of control over the model is a reason for exclusion. A change of LLM forced by the provider or operator is unacceptable from a business perspective, as the end of a life cycle is often announced at short notice and can cause considerable internal costs. The only viable option here is an open-source model, i.e., Llama 3 70B Instruct, which we can easily run on a hyperscaler. However, it is important that the model is deployed from our own download.

In scenario B, several options are imaginable. The model control isn't of much consequence here, and the providers' data processing agreements are sufficient. The obvious choice here would be to select the best model in a cost-performance comparison. According to the evaluation above, GPT-4o, Gemini 1.5 Pro, and Llama 3 70B Instruct are the most appealing models.

However, we can also take a different approach to the question of selecting a model. As mentioned above, competition between providers currently is fierce, and every month a different model is in the lead. Determining the most powerful model should therefore not be a one-off measure. As Meta's LLM shows, the step from one model to the next can entail an extraordinary quality leap, while in other cases the increase in performance is negligible. With continuous benchmarking, we can follow these market developments and decide objectively whether

This raises the question of which is the ideal hosting provider. Ideally, it should be able to provide most models and make changing the LLM as simple and cost-effective as possible. As the table above shows, AWS not only has the most models on offer. With the AWS Bedrock platform, it is also possible to change the LLM with little effort. In addition, AWS can claim a certain neutrality between the different models, as the Amazon LLM is not competitive (at least for the time being).

Making an informed choice with benchmarking

When selecting the right LLM, countless factors flow into the decision-making process. The non-functional aspects in particular require careful consideration and a high degree of finesse in order to factor in all relevant aspects and conditions. What is the use case? How is the IT infrastructure set up? Is data protection guaranteed? What are the available capacities of IT staff, and how big is the budget? These are just a few of the considerations that companies and organizations need to take into account before introducing an LLM.

Fortunately, evaluating the functional criteria is much less complicated with the LLM benchmarking we have developed. Our continuous assessment of existing and new LLMs provides you with a valuable tool for choosing the ideal option for your purposes and recognizing when it is financially and operationally worthwhile to switch to a new (competitor) model.

The results presented here are only a small part of our benchmarking. And just as the technology of LLMs is developing rapidly, we are constantly optimizing and expanding our test procedures. We are constantly incorporating new models into our assessment and plan to integrate further languages into our benchmarking, above all French and Italian.

new offerings, and our own application of large language models, we have the experience and expertise to ensure a successful long-term LLM strategy, even in a turbulent market situation.

Would you like to find out more about our benchmarking results and the introduction of large language models?

Contact a specialist

Published June 12, 2024

Share

Digital Innovation | Machine Learning (ML) | Conversational AI

Written by

Hartmut Keil

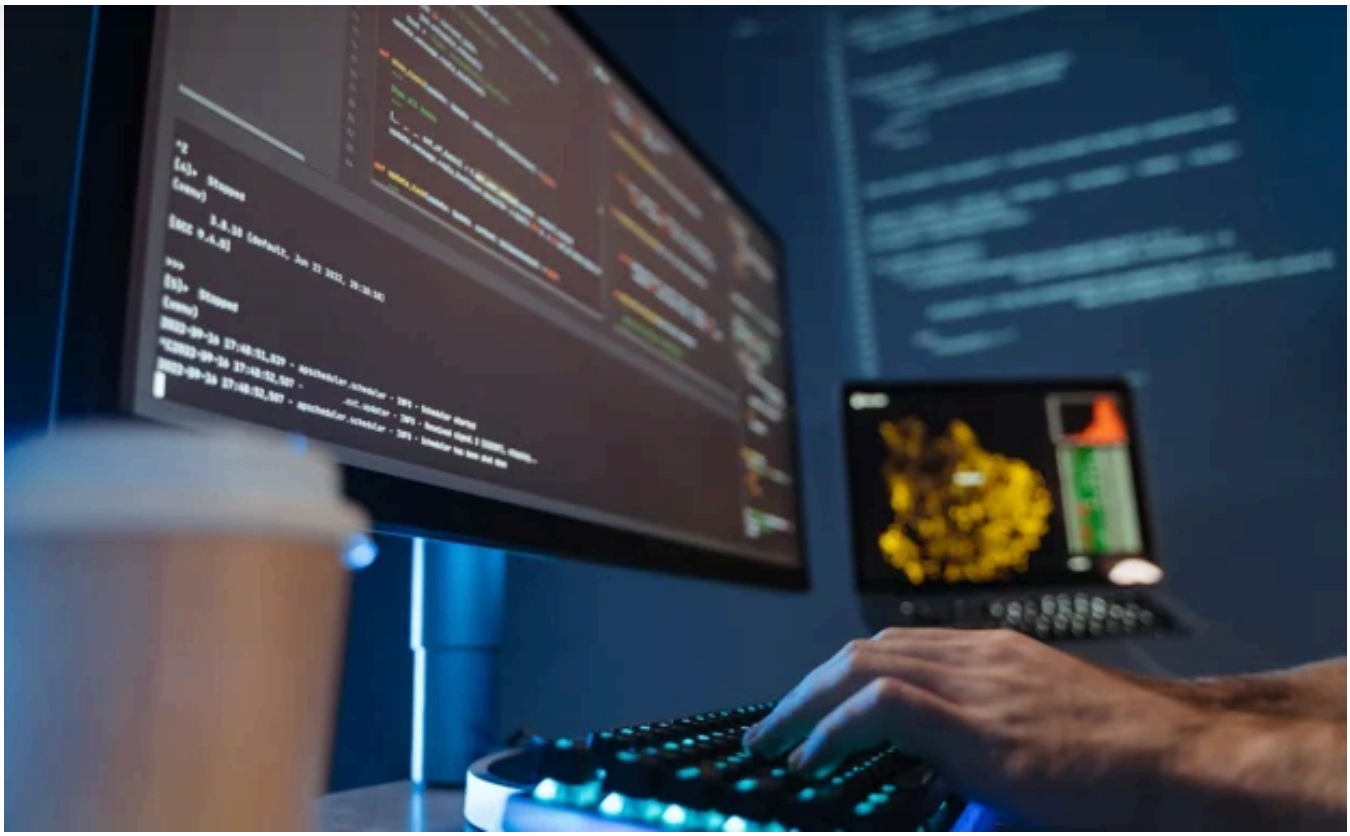
Principal Software Engineer

Go to our blog

Consulting

CIAM: Orchestrating a Secure and Frictionless Cloud Journey for Exponential Growth

13 min read



Consulting

AI in Cloud Security: Revolutionizing Defense Against Cyber Threats

13 min read

«There are different opinions about AI in software engineering»

13 min read

Ready to transform your business?

Let us know what your challenge is.

Talk to a specialist

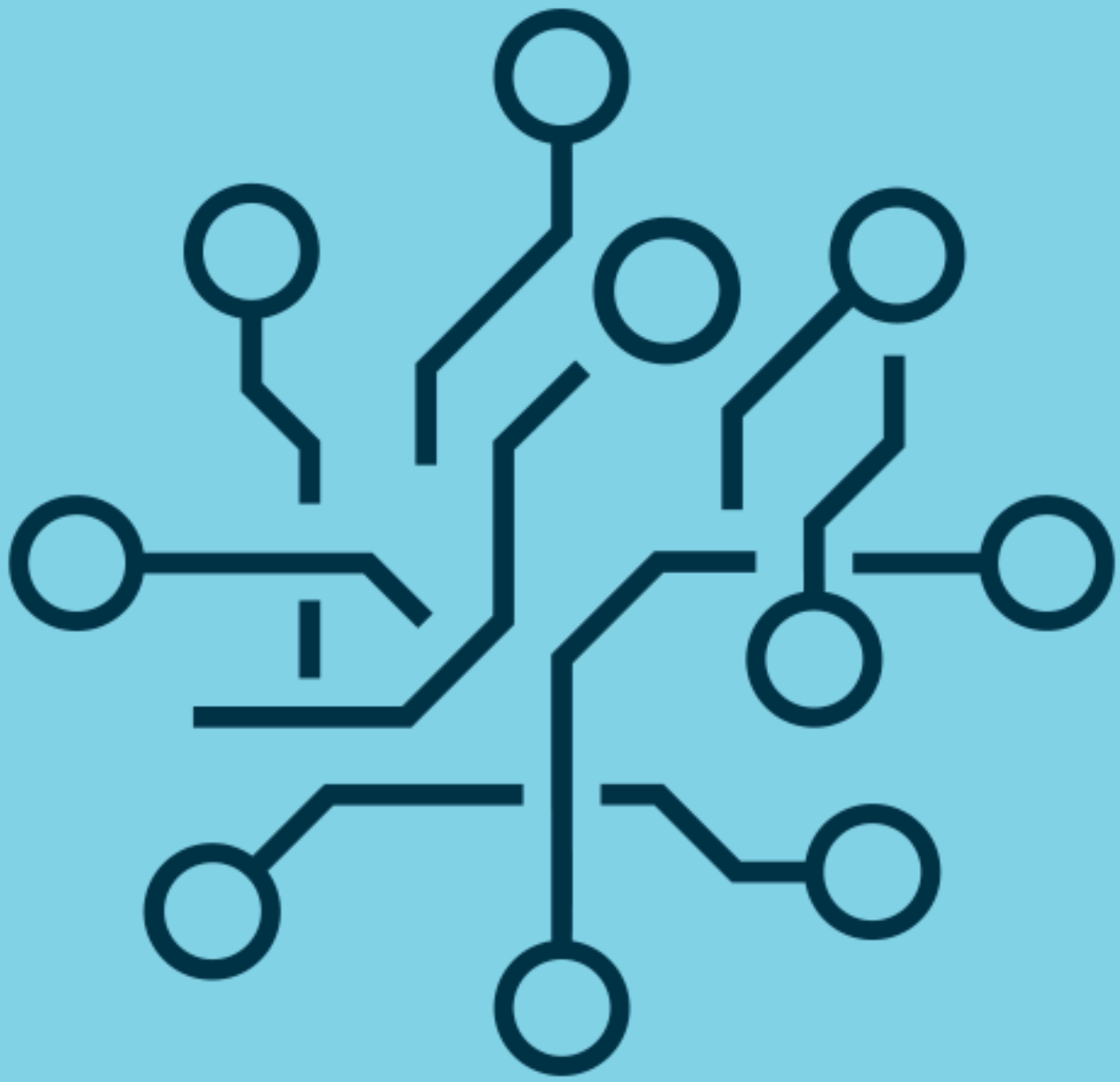
Stay updated

Subscribe to our newsletter to stay informed

Country

Subscribe

With respect for your privacy.



Company

Careers WE'RE HIRING!

Clients

Blog

Media

Contact and Location

Discover

Digital Solutions

Cybersecurity

Digital Innovation

Consulting

Application and Cloud Services

Industries

Banking and FinTech

Insurance and InsurTech

Public Sector

Transportation and Logistics

And your digital business works

Why Adnovum →

+41 44 272 61 11

Register for newsletter

Highlight

Digital identity: The Complete Guide to Digital Identification

Read the blog →