ScienceIO
by PM

Machine Learning   Large Language Models

Details

Review Date	12/11/2023
Purchase Date	Q3'21
Implementation Time	< 1 week
Product Still in Use	Yes
Purchase Amount	$8k / month for 100k tokens per day
Intent to Renew	100%
Review Source	Vendor

Product Rating

Product Overall	▬▬▬▬	4.5
Use Case Fit	▬▬▬▬▬	5.0
Ease of Use	▬▬▬	3.0
API	▬▬▬▬	4.5
Integrations	▬▬▬	3.0
Support	▬▬▬▬▬	5.0
Value	▬▬▬▬▬	5.0

About the Reviewer

N/A

Reviewer Organization

N/A

Reviewer Tech Stack

N/A

Other Products Considered

N/A

Summary

- **Product Usage: The product was used to analyze biotech news, specifically to accurately label company characteristics, drugs, and medical compounds involved in FDA approvals or rejections which aids in making trading decisions.**

- **Strengths: High accuracy in extracting and labeling medical information in text and the provision of relevant context which is essential for trading decisions.**

- **Weaknesses: The product's latency and over-labeling were initially problematic, and users must create their own tooling to cater the product to their specific needs.**

- **Overall Judgment: The product greatly aids in parsing medical text but can be improved if pre-built scaffolding is included for users, easing the process of application development.**

Today, we're chatting about ScienceIO and how it's being used at your company. Before we jump into that, could you give us a brief overview of the company and your role there?

I started a small fund during the COVID-19 pandemic. With my background in biology, I saw an opportunity to apply my knowledge to biotech stocks, so I started a small fund using family and friends' capital and started building the strategies. As the company grew, I hired a small team to work with me.

What was the need that drove you to look for a product like ScienceIO?

Initially, we used natural language processing (NLP) to analyze biotech news. We categorized the news into different types, such as FDA approvals, rejections, and mergers. The effects of that news on company stocks would vary depending on various factors, such as the size of the company and the significance of the event. For example, a small company getting an Alzheimer's drug approved—addressing a problem that hasn't been solved yet and for which there's a large market—would have a bigger impact than a large company getting a new ibuprofen alternative approved.

ScienceIO became crucial for us because it allowed us to quickly and accurately label company characteristics and pipelines as well as the drugs and compounds involved in drug approvals or rejections. We could take a press release and use ScienceIO to automatically identify the relevant medical compounds without any manual review. This allowed us to perform sentiment analysis, determining whether the news was positive or negative.

The main use case was for ScienceIO to parse out specific medical and drug-related terms into an ontology. We'd initially tried to build our own in-house solution, but categorizing medical terms proved challenging. We invested a lot of time and money in developing a very large model, but it wasn't effective. We also explored open-source models, including one developed by the University of Michigan, but none compared to the labeling accuracy of ScienceIO, which is vital when you're basing trading decisions on the data.

Did ScienceIO also handle sentiment analysis for you, or were you using other tools for that?

We used another tool for that. In its response, ScienceIO gives you the token, the category, and the position, so we could use that to easily expand into sentiment analysis.

Did they provide an out-of-the-box ontology for mapping language? Or was it an off-the-shelf ontology, like SNOMED?

They provide the actual UMLS ID, the word that appeared in the text, and the related concepts. The term "oral cancer" could be related to the concepts of cancer and dentistry, as well as the broader concept of disease. They provide all of those labels, creating a concept tree that gives every relevant label in a hierarchical structure.

Did you evaluate multiple vendors before choosing ScienceIO?

We built a rigorous test where we hand-labeled a range of news articles based on our requirements and then ran various models against it.

We tried the Michigan model, other open-source models, and a Hugging Face model, but we found them all lacking. We also tried another model called the Snow model, but we didn't use it for very long. Then we tried ScienceIO, and it blew everything else out of the water in terms of accuracy. It was absolutely incredible.

Which factors did you use to compare the models' performance?

We were looking at several criteria. First, we wanted to assess their accuracy in picking out relevant information. Some models tended to either over-fit or under-fit by over-labeling or under-labeling text, so we wanted to find one that struck the right balance.

Second, we wanted to evaluate how well the models used the context of the text to identify relevant medical concepts. We didn't just want them to pick out everything—they needed to focus on what was truly important in the given context.

We also considered how much information the models provided. Did they just give us a UMLS ID that required further parsing, or did they offer relevant context that allowed us to assess the importance of the medical concept within the larger context? For example, in a clinical trial, we were primarily interested in the concepts directly related to the drug being tested rather than secondary concepts that weren't directly related to the drug at hand, so the model needed to provide the necessary context for us to programmatically determine the relevance of scientific concepts and how a particular concept fits into the overall text.

Were there any other performance comparisons, such as latency of inference?

Latency was the most significant factor for us, especially as we needed a fast turnaround. At first, few other firms were using NLP to pick biotech stocks, so latency wasn't a big concern. However, as more and more firms started integrating NLP, latency became a crucial consideration. Fortunately, I was able to have direct conversations with the team at ScienceIO about this issue. I explained our requirements and pinpointed a major latency bottleneck, which stemmed from HTTPS authentication on their API. They made the necessary modifications and improved the speed significantly.

What pricing models did other vendors offer, and how did they compare to ScienceIO?

ScienceIO was definitely the most expensive option. Running the open-source models ourselves would have cost no more than about $1,000 dollars a month. With ScienceIO, the cost of our usage varied based on the number of press releases, ranging from as little as $1,000 to as much as $20,000. The baseline cost of using ScienceIO was significantly higher than what it would have cost us to either run our own model or host an open-source model; it was definitely pricier.

I'm not sure whether other competitors are offering different pricing models for a similar service; I haven't looked into it recently. But at the time, ScienceIO was the only one offering a subscription model, while everything else required us to handle the operations ourselves.

What was your experience with the sales process with ScienceIO like?

The sales process was quite informal since it was a smaller company. It was relatively easy: I emailed them, and they responded with a short sales call. We discussed the API integration, and they sent over some documents. It wasn't a very formal process. I reviewed the documentation, integrated the API, and then had a few back-and-forth email exchanges with them to address the questions I had.

They were good at seeking feedback from me on where they were excelling and where they could improve. We would communicate that to their engineering team, and then they would update me when they'd implemented a new set of features.

What was the onboarding and setup process like?

I would assume it's much more robust now, but at the time, it was a very small company, and the process was very informal and proceeded mostly over email.

I would love to get a comprehensive breakdown of the ScienceIO product. What are the different aspects and features of the model, and how have you found the performance across its different capabilities?

We only used the Identify API from ScienceIO. We didn't use their Redact, Learn, or Structure APIs because they weren't relevant to our specific needs. We focused solely on publicly filed releases. We're no longer using that strategy in our find, but I still personally use the model for a separate side project.

I have a medical search engine where users can input their condition in natural language, and the engine generates a plain English summary of the condition along with relevant information. For example, if it's cancer, it would provide details on life expectancy and current treatments, including experimental ones. For conditions like balding, it presents statistics and genetic influences in a way that is accessible. I plan on eventually releasing this project to the public, and I find the Identify API very useful for that purpose.

I believe ScienceIO is a very good application of AI. While generative models are receiving a lot of attention, I believe ScienceIO is a much better example of the use case for AI, where we can take difficult real-world problems and address them programmatically. We've made significant progress in fields such as programming, where there is a very defined syntax that you can easily interface with. Medicine doesn't work that way, so the impact of AI and technology in medicine has been relatively limited so far compared to other sectors.

However, ScienceIO provides a fundamental programmatic base for building applications and unlocking new possibilities in medicine, like a medical search engine. I think of it less as a competitor to an OpenAI-style model—I think it would do generative tasks poorly—instead, what I find useful is the concept of taking real-world items and then building an AI model that can parse those items reliably, which is what ScienceIO is good at. Its strength lies in accurately extracting information that matters.

How have you found the accessibility of the ScienceIO platform in terms of its ontology and understanding the overall tree structure?

ScienceIO's API is relatively straightforward—you can make an API call and observe the labeled output. When it comes to fitting the API into a specific use case, it's just a matter of inputting text and observing the response to come up with different use cases.

In our specific use case, we realized that accurately labeling relevant information in a text could help us build out the medical implications of FDA approvals or rejections without the need for extensive pre-research.

ScienceIO provides a flexible approach where you can input a piece of text and parse out the relevant medical concepts. It doesn't push you towards a specific use case like trading; instead, it offers a wide range of possibilities. For example, it could be utilized in a medical search engine or in a telehealth company. In the latter scenario, if you have a telehealth transcript where patients are discussing their symptoms, accurately extracting important concepts from the conversation could help programmatically assess the efficacy of doctors and open up new use cases.

The flexibility of ScienceIO's API allows developers to use the labeled information in a variety of ways, making it valuable for a broad range of different products; you have to create tooling around it to incorporate it into a useful product.

What are the strengths and weaknesses of ScienceIO as a product?

I think ScienceIO would do well to add some scaffolding to their product that would allow users to take their base API and easily unlock new applications. While they are already strong at extracting important context from a given text, they could enhance their offering by providing pre-built scaffolding that would eliminate the need for users to build additional components themselves, allowing for easier and broader application of the service. For instance, in my medical search engine project, I had to create my own separate database and context. If ScienceIO provided a ready-to-use database that could be easily integrated, it would greatly streamline the development process.

What ScienceIO does exceptionally well is its core API service, which accurately provides medical concepts from text input. This is crucial because if the API itself performed poorly, even with excellent scaffolding, it would be of little use. Their core service offering is very good; I just think they could use some extra work around refining and enhancing the overall user experience, making it simpler to build and expand new ideas without having to construct an entire toolchain from scratch. In the same way as Apple, for instance, has built Xcode, it would be extremely useful if ScienceIO built something that works like Xcode for medical applications. Developers would be able to bootstrap medical-related applications more easily and efficiently; this would enable them to unleash the full potential of ScienceIO's capabilities and explore the various possibilities of integrating technology into real-life medical scenarios.

It's difficult to overestimate how important this technology could be, but I think they can do a better job of actually being able to share it.

In terms of stability, reliability, and bugs, how did you find the ScienceIO product?

Initially, there were problems with latency and with over-labeling. For instance, when testing with a simple example like oral cancer, the model would provide information related to oral hygiene that was not directly relevant to the context of the document. This has since been fixed, and the API has been quite stable for about a year now in terms of the range of output it provides.

Could you provide more insights into the developer experience with the documentation and overall structure of the ScienceIO API?

The API is structured logically, and the documentation is great – it's easy to get started. They also provide a useful playground where you can test and experiment with parsing documents.

I built their TypeScript SDK since they initially only had a Python SDK available. We were doing some work in TypeScript, so I ended up writing a TypeScript SDK on top of their HTTP API. It's been a positive developer experience so far.

Do you have any practical advice or tips for someone who may be considering using ScienceIO's API?

Elion

For whatever use case you have in mind, there will be some tooling that you'll have to build, so think beyond its basic functionality and consider how you can leverage it to add value to your specific use case. Keep in mind that it can be quite expensive, so exploring ways to optimize its usage is crucial. For example, you can consider caching the API responses since they are stable, allowing you to save on API calls and optimize performance. ScienceIO's API provides a bare-metal solution, so you have the freedom to build on it and customize it as needed while being mindful of cost implications.

Did ScienceIO offer a free trial for you to benchmark and evaluate the product?

I paid a fixed rate for a set number of tokens over a three-month period to get started, so there wasn't a free trial, but it wasn't very expensive.

What was your experience like working with the ScienceIO team in terms of support and account management?

It was excellent. I talked to the team all the time. I could email them about any issues, and they would respond quickly. For example, when I asked them to adjust the ML model to prevent overfitting, they made the changes in a reasonable timeframe. It was very personalized, not like dealing with a separate sales team. I could just talk to them directly.

Do you think you made the right decision to go with ScienceIO?

Absolutely – it helped us to accurately classify medical concepts quickly and accurately.

You mentioned earlier that one thing you would like is for ScienceIO to create the Xcode of life science. Could you elaborate on that?

In terms of building an app, what Apple does very well is the tooling around the app. The process of building an app with Xcode is efficient and allows for small app sizes that deliver significant value and a fluid experience. While it is possible to build these features on your own, Apple provides a higher level of quality and allows for faster app development.

As for ScienceIO, their ability to take any medical idea and convert it into a format that computers can understand is incredibly powerful—it opens up numerous use cases and possibilities, many of which have similar basic parameters. For instance, you need databases. You have concepts structured not just as a tree but as vectors, allowing you to see how different pieces of information connect to each other. You might want benchmarks or data science tooling to help you analyze the connections between multiple pieces of previously analyzed text. You can build a lot of things around that. You can build a database that caches ScienceIO responses and a custom model to represent specific pieces of text without having to pass it to their API again. You can build relevancy scores based on their responses and do your own text extraction. Taking their API and building a chatbot off it in real time is not difficult; they have a good embeddings API, and integrating it with ScienceIO's platform can greatly benefit users. You could arrange it so that users don't have to do all of this surrounding work, and users could spend less money on their API, querying it only when it's really needed.

I think it's one of the most promising companies of the decade, capable of ushering in a new era of growth in medicine. Currently, medicine is primarily analog, but ScienceIO has the potential to revolutionize it by visually representing complex medical concepts. For example, in genome research, their platform could speed up research and drug development by labeling concepts very well. There are several areas like that, where they could become the de facto platform.

Do you have any advice for someone who is considering purchasing a classification or regression model? Any thoughts or recommendations?

I think it's important to avoid falling into the common pattern we tend to fall into with APIs, especially when it comes to ScienceIO. Instead of treating it as just another information point that you plug into your existing system, it's better to see it as a platform that you can build on. By developing custom tools around it as quickly as possible, you can fully leverage its power.

Let's say you're an insurance company. Don't just use the API to label medical conditions to price insurance better. Instead, consider how you can fundamentally re-architect the pricing process by incorporating better prediction models and exploring correlations between different factors. ScienceIO's API provides a rich semantic signal that can help identify high-risk and low-risk individuals. Think about how you can use it as a platform rather than just an API.