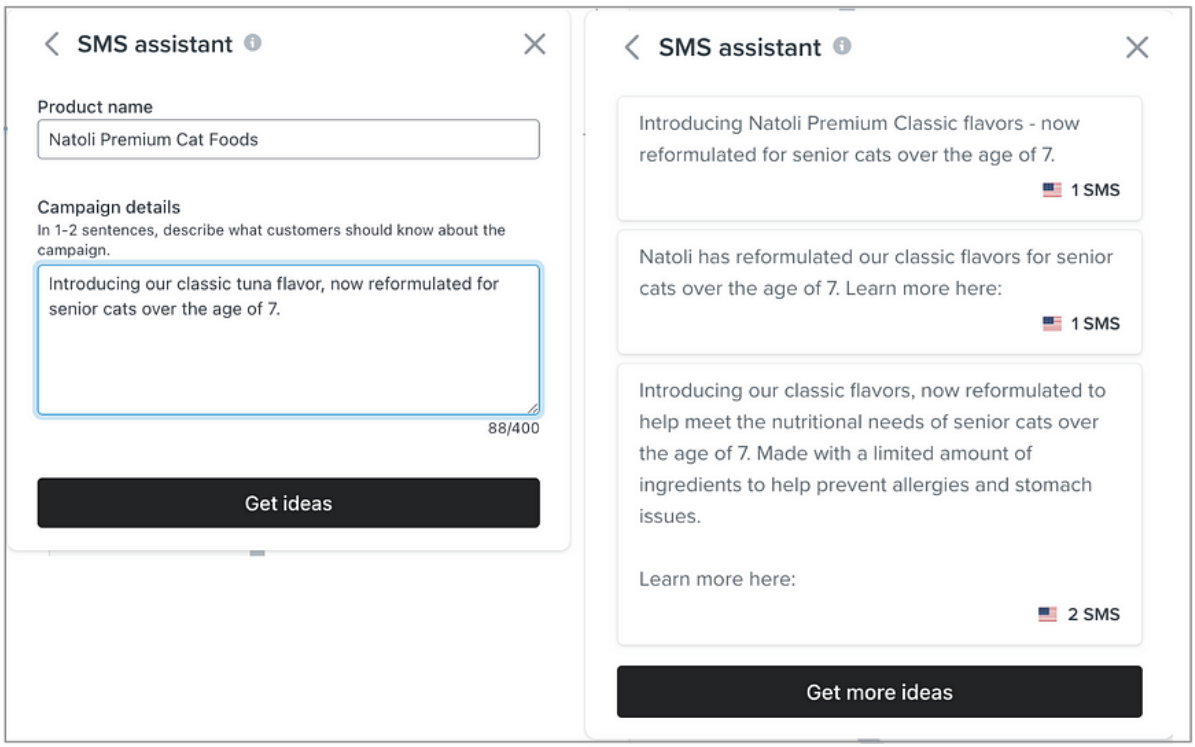
Lessons Learned Using GPT-3 to Build Klaviyoâ€™s SMS Assistant

Author: Charlie Natoli

Claps: 80

Date: Jan 24

Over the last year, Klaviyo launched tools that write creative email subject lines and SMS messages for our users. The way these work is simple â€” the user provides a description of their message (e.g. â€œweâ€™re having a sale on our scented candle collectionâ€�) and we return multiple marketing copy ideas. Under the hood, we use OpenAIâ€™s GPT-3 model.



Putting our SMS assistant to use for my *totally real* cat food business.

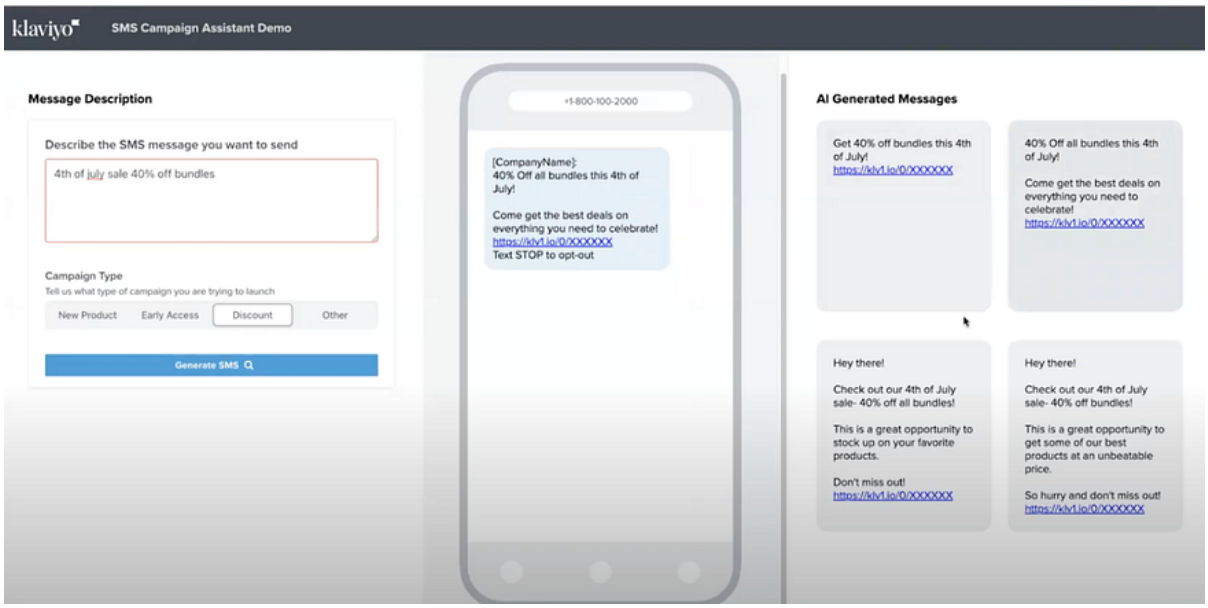
I was the data science lead on our SMS assistant. The project was different from other data science problems Iâ€™ve tackled in the past. First, working with generative AI models (whether generating text or images) is different than using models for other data science applications, such as classification or prediction. Second, while customer understanding is always key for a data scientist, I felt it was especially important here due to the subjective and open-ended nature of copywriting.

Here are lessons I learned.

When building a product using generative AI, actively participate in user research.

In many data science projects, itâ€™s often somewhat clearer what the goal of a model might be, for example classifying fraud, recommending products, or forecasting churn. However, for something as open ended and subjective as creative text generation, it was less clear what exactly the model should be doing. For example, are there certain marketing best practices to follow? Do users want you to closely follow their prompt? Or do they want you to show them new, creative ideas?

To get a sense for this, it was critical to work with the product manager and designer on the project in interviewing customers. We started by asking customers about their creative process and what makes it hard. However, just asking in the abstract often wasnâ€™t enough; the best way to get specific and actionable feedback was to show users real examples and let them play around with a prototype. To do this, we built a lightweight app that allowed users to play with the model, and used it in our conversations.



A screenshot of the tool we built to demo the SMS Assistant in interviews. It proved to be a great conversation guide.

From these interviews, we were able to get at more granular requirements for how the model should behave:

- **Be creative.** Many users, particularly small business owners wearing multiple hats, were constantly struggling to come up with new and creative ways of introducing things like product releases or sales. This suggested to us that taking creative leaps from the userâ€™s prompt was a key goal.
- **Avoid made up details.** Our model sometimes made up details that werenâ€™t true (such as discount codes that the user didnâ€™t specify in their prompt). While these seem like realistic details that might go in a promotional text message, they are unhelpful and (understandably) look weird to users. (There seems to be increasing public awareness of this problem, sometimes referred to as â€œhallucination,â€ especially given the examples of this happening with ChatGPT.)

- **Limit message length:** Users also cared a lot about having a mix of longer and shorter options. On the one hand, longer SMS messages cost more to send. One the other, it can be hard to convey your full point in a creative way within a tight character limit.

Performance of your generative AI model is hard to quantify before launching.

Before launching, quantitative metrics are tempting but often not worth it. As data scientists, weâ€™re always thinking about how to quantify things like model performance. In this project, I was tempted to do this here as well. Many academic papers test models against existing NLP benchmarks, and we wondered if we could do something similar. Doing so would help more formally compare different model versions, and be more reproducible than saying â€œyeah, we looked at some examples and we think itâ€™s good.â€

However, was there a benchmark that closely captured what we were trying to do? The answer for us was â€œnot really.â€ As described above, just figuring out what we wanted to optimize for was hard. Many existing benchmarks measure something much more narrow like sentiment, formality, or politeness, whereas our goal was a lot harder to define. In the end, meticulously looking through examples and validating that they were good enough based on customer learnings gave us confidence to move forward.

Test things live, running experiments where you can â€” thereâ€™s no better way to learn than that.

Even with deep customer research and testing, itâ€™s still hard to say exactly how well your model will fare with customers, for a few reasons:

- Users may react differently in reality compared to how they react in interviews. For example, customer interviewees may shy away from giving you negative feedback out of politeness, or they canâ€™t tell how useful the tool is until theyâ€™re actually sitting down to write a promotional SMS.
- There may be important edge cases (in our case, types of prompts people give to the SMS Assistant) that you didnâ€™t think of going in. Your model may not be robust to all of these.

In some cases (for example, a medical chatbot), there may be severe consequences to a bad message being generated. However, for a creative/marketing use case, this wasnâ€™t a problem as long as there were enough good answers that the user got value from the tool. So our aim became to get the model into the hands of real users. But, to test and tweak how our model fared, we ran a longer than normal beta period. We also considered (and continue to look at):

- **Message selection rate**. Of everyone who used the tool, what percent clicked on at least one of the messages generated? What factors were correlated with higher selection rates?
- **Use retention rate**. What percent of users came back and used the tool again a second time? What factors were correlated with higher retention?
- **Message quality.** We also looked at a large sample of both prompts and messages, to get a sense for what types of things people do and donâ€™t put in. Here, we saw a much wider

range of inputs than we expected, from some users putting in only a word or two, to others pasting in large amounts of text from other marketing copy.
- **Message filtering rate**. While we only need to return three messages to the user, we generate more than we need, and then filter some out that donâ€™t meet various criteria we have set. For example, messages that are too long, contain toxic content, or contain made up discount codes all get filtered and not shown to the user. Sometimes, we arenâ€™t able to generate three good ones. To ensure long-term success, we studied if certain types of prompts were more likely to struggle generating enough acceptable responses.

Wrapping up

Overall, working with generative AI has been a fantastic challenge, and a very different type of challenge than traditional data science work. These technologies will likely get integrated into lots of types of software. This seems even more clear after the recent release of ChatGPT which has raised awareness of generative AI and is changing user expectations for whatâ€™s possible.

For all data scientists interested, Iâ€™d encourage you to read up on how the models work, and what training data was used. Read how others online have tested GPT-3 (and increasingly, ChatGPT). Push your teams to consider both the range of possible opportunities, as well as risks, in using these models. And most importantly, Iâ€™d encourage you to play around with the models yourself.

Links

NLP benchmarks: This paper gives a good review of some of these benchmarks, as well as their pros and cons compared to human evaluation.

Exploring what GPT-3 can do:

- See our recent blog post exploring the range of GPT-3â€™s cousin, ChatGPT can do.
- Another great and really thorough account Iâ€™ve found of what GPT-3 can do are blog posts from Gwern Branwen on nonfiction and fiction.

Possible harms from large language models: I found this paper to give a really helpful overview of potential future harms from large language models overall. Some of these apply directly to our case of generating marketing copy.