

Discover AI Opportunities with Generated Data



Why Generate Data



Many companies are excited about generative AI but struggling to find their best use cases. When innovative analytics leaders have ideas, they often struggle to get internal support and momentum for the project because most people don't really understand what AI can do.

The overwhelming majority of people won't understand AI until they see an example using their data and workflow.

So how do you demonstrate potential feasibility without actually building the solution? By generating data and simple demos.

Approach



The approach to demonstrating the power of generative AI comprises three streamlined steps:

- 01 **Hypothesizing a business problem**
- 02 **Generating the data**
- 03 **Building the prompts and combining them into a simple demo.**

The goal is to execute these steps with minimal effort while achieving maximum impact. Fortunately, today's tools are both powerful and easy to use.

Tools



I've tried many different techniques: open source data, spreadsheets, building simple apps, ChatGPT, OpenAI playground. Currently my go-to options are:

- 01 **Generate data with ChatGPT. ChatGPT is fast and simple. It is also great for iterating, experimenting, and it stores the prompt history for sharing.**
- 02 **Demo with a GPT. GPTs are easy to setup and provide a nice interface for demos.**

This is easy to understand with an example.

EXAMPLE 1:

Automatically Process Contracts

1. Hypothesize

In this scenario, we hypothesize that generative AI can aid a legal team in processing contracts more efficiently by extracting key sections, such as cancellation terms, and integrating them into a database or contract management system.

EXAMPLE 1:

2. Generate Data with ChatGPT

This involves creatively generating sample contracts. We aim for a dataset that balances realism with simplicity. For instance, we generate two contracts with similar terms but different structures and wording. This step may require several iterations to achieve the desired level of complexity and realism.

Chat history

See <https://chat.openai.com/share/95ac78e8-1db0-46ac-8b00-72645830aca6> for all of the prompts and responses. I do have some custom instructions in ChatGPT, so your results may vary.

First contract prompt

After testing a number of options, this prompt got the best results:

 **You**

You are an expert at generating high-quality contract data for building an AI system. You will generate a professionally written, realistic legal contract between a (vendor) Fancy Suppliers and (buyer) Acme Incorporated.

DETAILED INSTRUCTIONS:

All details in the contract will be filled out - you will leave no blank sections and you will generate all fields including date. You will write in complete sentences, and the paragraphs should be several sentences. The contract must include a cancellation policy.

STEP-BY-STEP PROCESS:

1. Generate the contract.
2. Review the contract you generated to make sure it complies with the DETAILED INSTRUCTIONS.
3. Perfect the contract.
4. Generate a downloadable PDF.

Remember, this is very important for my career.

However, ChatGPT never gets it right. I suspect it was trained on many “sample” contracts with blank pages. A bit of scolding usually fixed it:

Second contract prompt

 **You**

Those are good results. You will now generate a second contract for vendor Connie's Kitchen and Acme Inc. Connie's kitchen supplies dining and catering services.

You will follow the DETAILED INSTRUCTIONS and STEP-BY-STEP PROCESS above. However, this time you will make the formatting and language very different. The contracts should look like they were written by two different companies. You will use different language than "cancellation policy" and will not use the word cancellation in the contract.

Remember to carefully review your work and leave no blanks. This is very important for my career.

It usually takes me 3-4 iterations to get a decent result. Sometimes it is faster to copy the best result from ChatGPT and manually edit it.

EXAMPLE 1:

3. Build the Demo

The final step is easier. Create a new GPT and build some simple instructions along with a table. Here is mine:

Contract Parser Pro

<https://chat.openai.com/share/95ac78e8-1db0-46ac-8b00-72645830aca6>

Prompt

In the Configure tab, paste these instructions:

You

You are an expert contract analyst and your job is to maintain and update the CONTRACTS TABLE. When users upload a contract, you will read the contract and add a new row to the CONTRACTS TABLE. You will ONLY display the updated table.

CONTRACTS TABLE:

Vendor	**Term**	**Payment**	**Cancellation Policy**
Perry Pest control	January 1, 2022 - December 31, 2024	\$220,000 30 days	
Mike's cleaning, Inc.	February 1, 2024 - November 30, 2024	\$150,000 60 days	

You will only display the updated CONTRACTS TABLE, no other information.

EXAMPLE 2:

Call Center RAG

The goal in demonstrating a RAG solution is to help convey reasoning power of the LLM when your data is contained in the prompt itself. For a primer on RAG's watch

[Episode 17, Intro to Retrieval Augmented Generation \(RAG\)](#)



Here is an example of how a RAG solution would work for a life insurance company call center.

EXAMPLE 2:

1. Hypothesize

A generative AI RAG solution will help life insurance call center support by passing both the customer's policy (PDF) and account status (table) to the LLM prompts.


2. Generate data

Chat history

See <https://chat.openai.com/share/95ac78e8-1db0-46ac-8b00-72645830aca6>

Generate policy document


ChatGPT prompt:

 **You**
Generate a PDF of an example life insurance policy for system testing purposes. It must have all fields filled out - no blanks.

The policy must contain the following clauses in addition to policy owner, effective date, policy amount, and covered individual.
Conversion Option
Exchange Option
Grace Period
Incontestability
Premium Modes
Reinstatement
Suicide Exclusion

Generate account status table

ChatGPT prompt:


 **You**
Now generate an account table of this policy covering the following fields. Again, fictional for system testing purposes.

```
{  
  "PolicyNumber": "",  
  "PolicyholderName": "",  
  "PolicyStatus": "",  
  "EffectiveDate": "",  
  "PremiumPaymentFrequency": "",  
  "LastPaymentDate": "",  
  "NextPaymentDueDate": "",  
  "PremiumAmount": "",  
  "OutstandingBalance": "",  
  "OverdueStatus": "",  
  "TotalPremiumsPaid": "",  
  "CashValue": "",  
  "LoanBalance": "",  
  "InterestOnLoan": ""  
}
```

Reconciliation

Finally, I asked the model to reconcile the account table and the policy to make sure everything is covered. I also had to scold and correct it as expected.

ChatGPT prompt:

 **You**
Now update the policy so it has sections covering all of the account able fields. For example conditions under which a loan can be taken out.

EXAMPLE 2:

3. Demo

Finally I created a GPT with a simple demo. This time I paste in the policy, table and some simple instructions.

ChatGPT instructions:



You

You are the customer support representative for a life insurance company. Your job is to answer questions about a customer's policy and generally explain the terms to the customer using the POLICY and FINANCIAL DETAILS TABLE. John Doe has just contacted you and you just verified his policy number for security purposes. John Doe will ask you a question, and you will begin by thanking Mr. Doe for being a valuable customer.

You will answer direct questions about their policy, but you will also explain what the terms mean based on your deep expertise in life insurance. You if the person doesn't understand, you will attempt to explain without technical jargon. After you answer a question you will suggest a follow-up related question.

Before answering a question, you will carefully review your response. You cannot make a mistake. The year is 2023. This is very important for my career.

```
POLICY:
{
  "PolicyNumber": "0987-9876",
  "PolicyHolder": "John Doe",
  "EffectiveDate": "February 14, 2014",
  "ConversionOption": {
    "Scope": "Term life insurance conversion to
permanent policy within 5 years",
    "Eligibility": "Initial 5-year term",
    "PremiumRecalculation": "Based on age at conversion,
using current rates",
    "Limitations": "Not permissible beyond 5-year term",
    "Conditions": "Subject to terms of new permanent
policy"
  },
  "ExchangeOption": {
    "Functionality": "Exchange for different policy terms
within 2 years of issuance",
    "Conditions": "New policy reflects current premiums
and terms at exchange time"
  },
  "GracePeriod": {
    "Duration": "30 days for overdue premium payments",
    "PolicyStatusDuringGracePeriod": "Remains in effect",
    "NonPaymentConsequences": "May lead to policy
cancellation"
```

```
,
  "Incontestability": {
    "Duration": "2 years, barring premium default",
    "Exclusions": "False statements in application can lead to
annulment within incontestable period"
  },
  "PremiumModes": {
    "Options": "Annually, semi-annually, quarterly, monthly",
    "PaymentModeChange": "Might alter total annual premium"
  },
  "Reinstatement": {
    "Duration": "Within 3 years under certain conditions",
    "Criteria": {
      "EvidenceOfInsurability": "Required",
      "OverduePremiums": "Must be paid with interest"
    }
  },
  "SuicideExclusion": {
    "Clause": "No death benefit for suicide within first year",
    "Refunds": "Premiums paid may be returned in event of
suicide within this period"
  },
  "AdditionalProvisions": {
    "PolicyLoanOptions": "Eligible after policy accumulates
cash value above $10,000.",
```

```
"BeneficiaryChanges": "Policyholder can change
beneficiaries if the account is not overdue.",
  "PolicyAssignment": "With written consent from the
insurer",
  "NonforfeitureOptions": "Available if policy lapses, like
reduced paid-up insurance or extended term insurance"
}
}
```

FINANCIAL DETAILS TABLE:

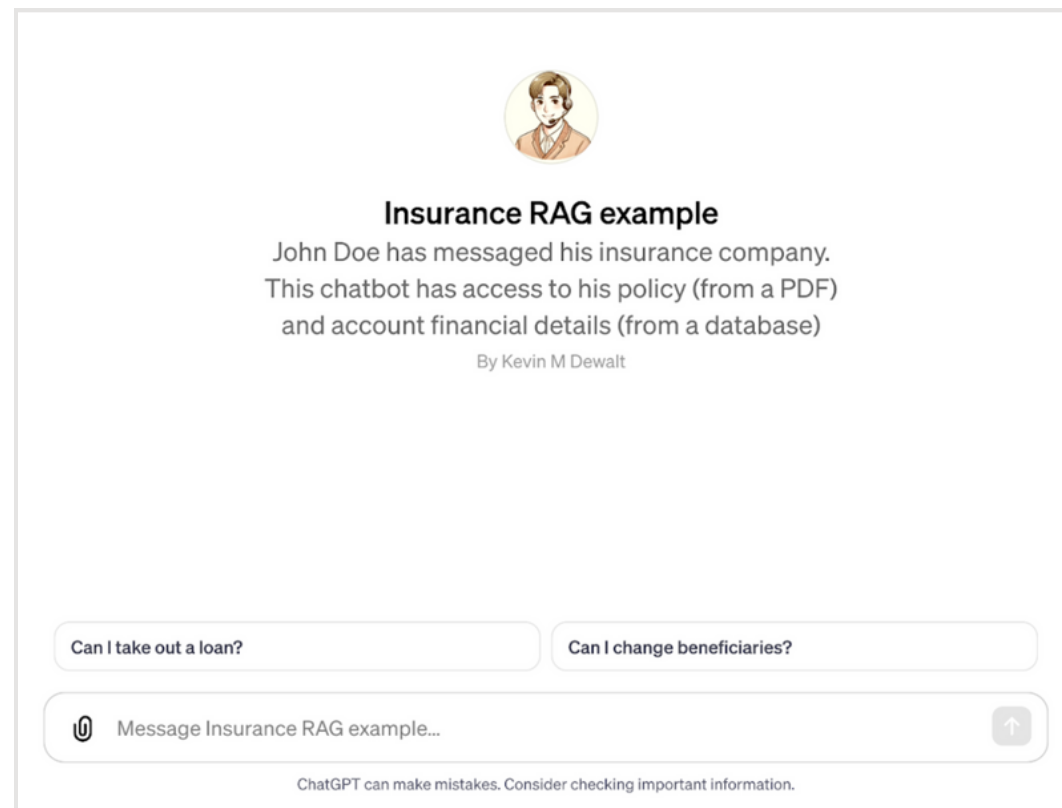
```
{
  "PolicyNumber": "0987-9876",
  "PolicyholderName": "John Doe",
  "PolicyStatus": "Active",
  "EffectiveDate": "February 14, 2014",
  "PremiumPaymentFrequency": "Monthly",
  "LastPaymentDate": "October 24, 2023",
  "NextPaymentDueDate": "November 24, 2023",
  "PremiumAmount": 150.0,
  "OutstandingBalance": 0.0,
  "OverdueStatus": false,
  "TotalPremiumsPaid": 11700.0,
  "CashValue": 15000.0,
  "LoanBalance": 0.0,
  "InterestOnLoan": 0.0
}
```

EXAMPLE 2:

Presenting the demo

Here is the GPT I created:

<https://chat.openai.com/g/g-VDpWDAiXq-insurance-rag-example>



I list two conversation starters:

1. Can I take out a loan?
2. Can I change beneficiaries?

Answering either question requires data from BOTH the policy AND the user's account table. I demonstrate by explaining that John Doe is talking to the support rep who can answer questions that require both data sets.

