

OCT 22, 2024

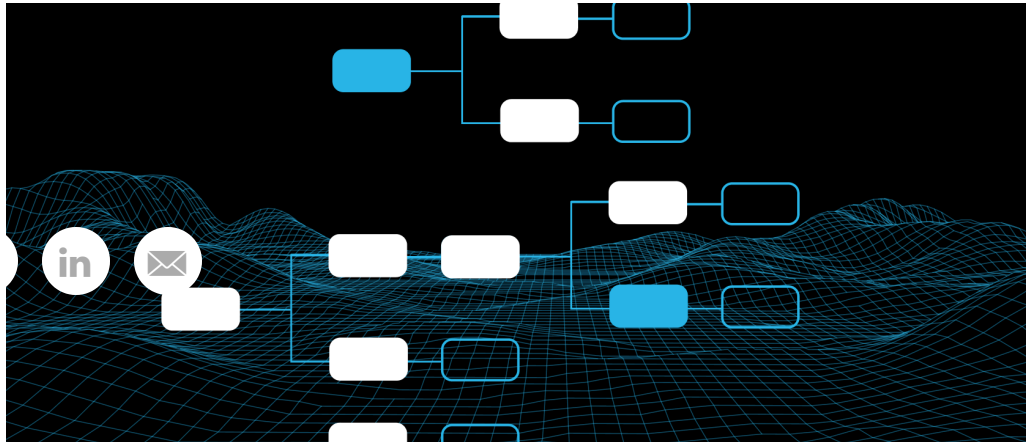
AUTHOR



Nikolai Scholz

Best Practices for Getting Started with Snowflake's Document AI

Gen AI



SHARE

Transforming unstructured data to value

During the preview of Snowflake's Document AI, we engaged with hundreds of customers eager to empower their business teams with automated intelligent document processing (IDP) workflows. It all begins by unlocking the data that is buried within unstructured files, ultimately resulting in the democratization of this extremely valuable source.

Document AI brings a new paradigm to Snowflake's suite of functionalities by putting documents and images on the same level as data tables, bridging the gap between structured and unstructured data.

Today, we are proud to announce the general availability (GA) of Snowflake Document AI, powered by our proprietary Arctic-TILT model, on Azure and AWS. Document AI transforms unstructured documents into structured, actionable data, seamlessly integrating it into your Snowflake data platform. By leveraging generative AI, Document AI drastically reduces the amount of time required for data extraction, enabling users to parse both structured and unstructured documents quickly and accurately — with minimal or no training. Additionally, developers have no need to manage complex tasks like converting documents to text, managing document chunks or optimizing extraction prompts. Document AI streamlines these processes end to end, allowing

for more efficient extraction pipeline creation. With flexible model usage — including out-of-the-box (zero-shot) capabilities, fine-tuning options and support for document classification — this enterprise-ready solution offers reliability and scalability for production use. You can [watch the demo here](#) to learn more about how Document AI works.

In this post, we summarize our learnings and provide best practices for leveraging the technology to optimize document management and processing within Snowflake.

Before getting started

Explore broadly: Focusing on Business Documents, speak with your business teams and consider the full array of potential use cases. Examples include: invoices, receipts, contracts, agreements, identity documents, lab reports, surveys and many more.

Begin with familiar documents: The more familiar you are with the initial documents you use, the better you'll be able to evaluate whether the results are accurate. If possible, involve your subject matter expert early on.

Data availability: Ensure there is enough suitable training and evaluation data for your use case. It is important to have enough of both to evaluate the model's ability to solve the use case.

Aim to reduce manual work and errors: One of the primary benefits of using Document AI is reducing human labor and errors. With this in mind, having an understanding of current playbooks, SLA's and inefficiencies will help in assessing the value quicker. Even partial solutions can have a significant impact on efficiency and reduction in extraction error rates (i.e., reducing manual labor by 50%).

Figure 1. Do's and Don'ts with Document AI: Make sure to always check the Document AI documentation for the most recent updates on limitations.





Best practices for using Document AI

Building models

One vs. many: Consider each model build as a solution tailored to your specific use case. Ideally, a single model should address the entire range and diversity of the document types involved. In only rare cases would you need more than one model build for the same document type.

Involve your SME/document owner: Your subject matter experts and document owners are crucial partners in understanding and evaluating the model's effectiveness in extracting the required information. Incorporate existing playbooks and output tables (i.e., existing Excel sheets with past extractions) into the evaluation process.

Defining data values with questions

Be simple, yet precise with data values: Questions should be clear and     a specification where appropriate, but avoid reliance on domain knowledge that is not present in the document. The model should be able to repeatedly find the answer. If that is not the case, then either:

The question is not encompassing or specific enough.

There are documents with variations in layout or nomenclature, requiring further training for the model.

Fine-tuning over prompt engineering: Experimenting with different prompts can be a time-consuming process, often leading to a variety of initial responses that need further refinement. This is especially true given the complexity and variability of documents, which means that initial responses may not always be precise or fully appropriate without additional fine-tuning. It is recommended to keep the question as encompassing as possible and pursue training with annotations.

Training and evaluation

Good data representation: Your training data — or more specifically, the corrections and annotations that you accept in order to train the model — should include a fair and balanced representation of value occurrences across your general body of documents. This includes: *Layout*, *Nulls*, *Lists*, *Class Representation* (in case of a classification task) and *Answer Variety* (showing range and variety).

AUTHOR



Nikolai Scholz

SHARE

Training helps with the model's confidence: The confidence score measures the model's certitude that its answer to a given question is correct. Just like any confidence, it can be baseless. If you notice false positives, or even false negatives, training with additional cases and corrections will help.

Evaluation of the model in development and production: As is recommended across any automated system, it is important to set evaluation criteria and methods to observe the processing. It is recommended to create an independent evaluation set of documents and individual confidence thresholds on a data-value level to be able to track and identify results in need of additional reviews. Make sure that any evaluation data does not include data that was used in training.

Tips and tricks

You do not have to annotate all documents; a smaller training up front
f nā' in su' ✉ uent annotation easier, if deemed necessary.

When first assessing a new model build, take note of the corrections you are making on the first few documents. If these corrections are repetitive, such as date formatting, removing or adding prefixes or suffixes or other formatting changes, consider training beyond five documents. Doing this early saves time, as you will have a new starting point on Document 6, and require less annotation work.

Show, don't tell.

There are questions and values that will not be possible to answer or produce in zero-shot (foundational model without training). These might include: combinations of values, arrays, nonstandard formats, normalization and classification tasks. In such cases it is best not to spend much time on the question prompt itself but instead show the expected result in the form of annotations across the appropriate amount of documents in the training set.

Normalize results through annotations and training.

Normalizing results, such as date formats, locations, currency and numbers, is possible through training a model built on enough documents and annotations. The best practice would be to extract values such as numbers in the rawest form, as formatting can then be controlled when defining the data type.

AUTHOR



Nikolai Scholz

SHARE

Make sure to train in all classes when performing a classification task.

To perform a classification task that can sort through a mixed repository of documents, you can train a single-defined value, such as: “What is the document type?” and provide every iteration of a possible classifier (i.e., service agreement, transportation agreement, order form, etc.). Additionally, you can add more defined values into a model build that can, in total, produce a certain outcome in a post-processing step.

AUTHOR



Nikolai Scholz

You can reconstruct tables accurately using columnar extraction or training the model to extract rows in delimited form.

In order to extract data from tables that span across many documents and reconstruct them into one schema, there are multiple approaches available. With Document AI it is possible to extract columns of data, in lists, or in delimited form. These lists can then be merged in the pipeline to reproduce the table. Another way can be training individual values to extract rows in delimited forms (i.e., commas or pipes). It is vital in both cases that enough data is used to train the model to include NULL values and maintain order. Also, be sure to consider the size of the table.



SHARE

Document AI can extract from non-English documents.

Document AI formally only supports the English language. That said, the model has seen great levels of success with other Latin alphabet-based languages. If for any reason you might have documents in languages, such as French, German or Spanish, make sure to train the model appropriately. When defining the questions, write them in English and use any native term when appropriate.

Through our preview, we've also learned a lot about how customers are using Document AI to innovate their businesses. In the next section, we will outline common customer use cases we've seen.

Use case: Invoice extraction and processing

Similar documents include financial statements, surveys and lab reports

Question examples:

Key and general information

1. What is the invoice number?
2. When was the invoice issued?
3. What are the payment terms?

Table-related information

1. What line items are listed?
2. What are the amounts for each line item?
3. What is the description, quantity, unit price and amount for each line item?

Best practices:



Invoices come from many different sources. Make sure to come up with a training set that is representative of the full body of them. Keep in mind that in some cases it is not realistic to train on every type of format – use your discretion. Treat Document AI like a new employee and ask yourself: *Is the set of documents in training enough to infer and process unseen documents?*

When working with **Questions 4 and 5**, make sure to train the model to pull line items in the right order and represent *NULLs* and *Blanks* correctly. This means annotation and training on at least 20 documents.

Do not assume the foundational model will immediately understand the intent of **Question 6**. More than likely, the model will not get this one 100% correct the first time around. With training it will learn and understand from your desired output, through the annotations/corrections and training.

There are multiple ways to work with JSON lists when the target is a restructured table. Work backwards from your required end-schema to find the most suitable solution. Sometimes, an array is better suited

AUTHOR



Nikolai Scholz

SHARE

than multiple lists of columns. You can end up with two tables — one with general information (single-value output) and the other with line items, with the primary key being the invoice number — or you can have one table with all the information. Make sure that the tables are all within the current limits. If you believe the output could exceed the limit, then consider flagging such documents for additional reviews.

AUTHOR



Nikolai Scholz


Use case: Doctor's note and handwritten text

Similar documents include intake forms, letters and correspondences

Question examples:

Key and general information

1. What clinic does this note come from?

- f 2. in , at ,  the address of the clinic?

3. What is the phone number of the clinic?

Handwriting-specific information

1. Is there handwriting present?
2. What date was the note written?
3. Is the document signed?
4. Who signed the document?

Best practices:

The accuracy of extracting information from handwriting is highly dependent on the feature's ability to recognize the text. Depending on the readability of the document, the model may extract incorrect information with high confidence. In such cases, review your

SHARE

document and see if there are any means to programmatically improve the quality of the image or PDF – such as utilizing only original documents or resizing the documents to standard formats.

Additionally, you can utilize the OCR score in the result of the feature function to flag documents with lower readability scores.

Values such as **Question 4** can be used to classify and sort documents.

It is a great piece of meta information that can be trained simply across your documents.

AUTHOR



Nikolai Scholz

SHARE



For **Question 6**, make sure to show the model in training what you expect and accept as a signed document. If there are multiple fields that need to be signed, explore both the option of a single-defined value for the entire document and multiple values for each signature field.

When tackling values such as **Question 7**, you can either train by classifying every known signature (as long as they are limited) or go down the route of extracting the written-out signature. With the first option it is vital to show every signature in variation within training. With the second option it is very likely that not every character can be identified, so use post-processing validation to evaluate.

Use case: Contract management and processing

Similar documents include service agreements, statement of work and order forms

Question examples:

Key and general information

1. What contract type is it?
2. Is the contract signed?
3. Is the contract still valid?

1. Who are the signing parties?
2. Is there an indemnification clause?
3. What are the termination terms?

Best practices:

AUTHOR



Nikolai Scholz

SHARE



Documents like contracts are highly specialized and require experience to fully grasp. It is highly recommended to include your SME, operational or specialized team member as part of the process. In some cases it should be these partners who prepare and evaluate the model for production implementation.

Question 1 is a classification task. You can use such a question to sort out your repository of mixed document types. Make sure to show every class in the training set and annotate/correct the response in the format you expect.

For contract management, a very common use case is to identify standard vs. nonstandard terms. While it is not possible to teach the model to identify such definitions at this level, it is possible to infer from a combination of questions, such as **Questions 5** and **6**. Given this information, you should be able to produce a list of flagged contracts for your SMEs, reducing the reading hours significantly.

To learn more about Document AI, [watch this demo](#) or [try it yourself today](#).

RELATED CONTENT

AUTHOR



SEP 12, 2024

LLM Interactive Workloads: Optimizing GPU Capacity for Interactive and Batch Workloads

At Snowflake, we offer a wide variety of LLM-powered features in Cortex AI, including Cortex LLM Functions, Snowflake Copilot and our recently released Cortex Analyst, now in public preview. While...

[Delve into the details](#)

SEP 05, 2024

Model Hotswapping: Optimizing AI Infrastructure and Enhancing LLM Efficiency

At Snowflake, we support customer AI workloads by offering a diverse range of open source...

[Full Details](#)

JUN 17, 2024

Snowflake Arctic Cookbook Series: Instruction- Tuning Arctic

On April 24, we released Snowflake Arctic with a key goal in mind: to be...

[Expand your knowledge](#)

READ GENERATIVE AI AND LLMS FOR DUMMIES

[GET YOUR COPY](#)

AUTHOR



Nikolai Scholz

SHARE



PLATFORM

- Cloud Data Platform
- Pricing
- Marketplace
- Security & Trust

SOLUTIONS

- Snowflake for Financial Services
- Snowflake for Advertising, Media, & Entertainment
- Snowflake for Retail & CPG
- Healthcare & Life Sciences Data Cloud
- Snowflake for Marketing Analytics

RESOURCES

- Resource Library
- Webinars
- Documentation
- Community
- Procurement
- Legal

EXPLORE

- Blog
- Trending
- Guides
- Developers

ABOUT

- About Snowflake
- Investor Relations
- Leadership & Board
- Snowflake Ventures
- Careers
- Contact

Sign up for Snowflake Communications

diana.shaw@snow United States

By submitting this form, I understand Snowflake will process my personal information in accordance with their **Privacy Notice**. Additionally, I consent to my information being shared with Event Partners in accordance with Snowflake's **Event Privacy Notice**. I understand I may withdraw my consent or update my preferences **here** at any time.

SUBSCRIBE NOW

[Privacy Notice](#) | [Site Terms](#) | [Cookie Settings](#) | [Do Not Share My Personal Information](#)

© 2024 Snowflake Inc. All Rights Reserved | If you'd rather not receive future emails from Snowflake, unsubscribe here or customize your communication preferences

