



Transparency Note: Form Recognizer

Updated 02/16/2022

Table of Contents

What is a Transparency Note?	3
Introduction to Form Recognizer	3
Form Recognizer terms	3
Example use cases for Form Recognizer	4
Considerations when choosing other use cases	4
Characteristics and limitations of Form Recognizer	5
Accuracy	5
Word-Level Accuracy	5
Entity-Level Accuracy	6
Best practices to improve custom model quality	6
Prebuilt model limitations	6
System limitations and best practices to improve system performance	6
Customer evaluation	7
Learn more about responsible AI	8
Learn more about Form Recognizer	8
Contact us	8
About this document	8

What is a Transparency Note?

An AI system includes not only the technology, but also the people who will use it, the people who will be affected by it, and the environment in which it is deployed. Creating a system that is fit for its intended purpose requires an understanding of how the technology works, what its capabilities and limitations are, and how to achieve the best performance. Microsoft's Transparency Notes are intended to help you understand how our AI technology works, the choices system owners can make that influence system performance and behavior, and the importance of thinking about the whole system, including the technology, the people, and the environment. You can use Transparency Notes when developing or deploying your own system, or share them with the people who will use or be affected by your system.

Microsoft's Transparency Notes are part of a broader effort at Microsoft to put our AI Principles into practice. To find out more, see the [Microsoft AI principles](#).

Introduction to Form Recognizer

[Form Recognizer](#) is accessed via a set of APIs, and allows developers to easily extract text, structure, and fields from their documents. It is composed of features like Read for text extraction, Layout and General Documents for structural insights and general key-values and entities such as names, places, and things, prebuilt models for specific document types like invoices, receipts, business cards, W2s, and IDs, and custom models for building models specific to your document types.

Form Recognizer supports one or more languages and locales for each of the features as listed in the [supported languages](#) article.

Form Recognizer terms

Term	Definition
Read	This feature extracts text lines, words and their locations from images and documents, along with other information such as detected languages.
Layout	This feature extracts text, selection marks, and table structure (the row and column numbers associated with the text). See Form Recognizer Layout .
General Documents	Analyze your documents and associate values to keys and entries to tables that it discovers. For more information, see Form Recognizer General Documents .
Prebuilt models	Prebuilt models are document-specific models for unique form types. These models don't require custom training before use. For example, the prebuilt invoice model extracts key fields from invoices. For more information, see Form Recognizer prebuilt invoice model .
Custom models	Form Recognizer allows you to train a custom model that is tailored to your forms and documents. This model extracts text, key/value pairs, selection marks, and table data. Custom models can be improved with human feedback by applying human review, updating the labels, and retraining the model using the API.
Confidence value	All Get Analysis Results operations return confidence values in the range between 0 and 1 for all extracted words and key-value mappings. This value represents the service's estimate of how many times it correctly extracts the word out of 100 or correctly maps the key-value pairs. For example, a word that's estimated to be extracted correctly 82% of the time will result in a confidence value of 0.82.

Example use cases for Form Recognizer

Form Recognizer includes features that enable customers from a variety of industries to extract data from their documents. The following scenarios are examples of appropriate use cases:

- **Accounts payable:** - A company increases the efficiency of its accounts payable clerks by using the prebuilt invoice model and custom forms to speed up invoice data entry with a human-in-the loop. The prebuilt invoice model can extract key fields, such as *Invoice Total* and *Shipping Address*.
- **Insurance form processing:** - A customer trains a model by using custom forms to extract a key/value pair in insurance forms, and then feeds the data to their business flow to improve the accuracy and efficiency of their process. For their unique forms, customers can build their own model that extracts key values by using custom forms. These extracted values then become actionable data for various workflows within their business.
- **Bank form processing:** - A bank uses the prebuilt ID model and custom forms to speed up the data entry for "know your customer" documentation, or to speed up data entry for a mortgage packet. If a bank requires their customers to submit personal identification as part of a process, the prebuilt ID model can extract key values, such as *Name* and *Document Number*, speeding up the overall time for data entry.
- **Robotic process automation (RPA):** - Content extraction for a wide variety of form types and fields is supported via custom forms. Many industries have technical form types that have distinct structures and key/value pairs. To extract this data and make it actionable, customers can use custom forms to train custom models and enable RPA scenarios.
- **Consumer behavior and market analysis:** - Using the prebuilt receipt model, customers can quickly extract key values, such as the merchant name and transaction total, from retail receipts. Customers can then use the extracted data to do consumer behavior analysis. For more information, see [Form Recognizer prebuilt receipt model](#).

Considerations when choosing other use cases

Consider the following factors when you choose a use case.

- **Carefully consider when using for awarding or denying of benefits - Medical insurance:** These include healthcare records and medical prescriptions that are a basis for decisions on insurance reward or denial. **Loan approvals:** These include applications for new loans or refinancing of existing ones.
- **Carefully consider applying human review when sensitive data is involved** - It is important to include a human-in-the-loop for a manual review when you're dealing with high-stakes or sensitive data. Machine learning models are not perfect. Consider carefully when to include a manual review step for certain workflows. For example, identity verification at a port of entry such as airports should include human oversight.
- **Carefully consider the supported document types and locales** - Prebuilt models have a pre-defined list of supported fields and are built for specific locales. Be sure to carefully check the officially supported locales and document types to ensure best results. For example, see [Form Recognizer prebuilt Receipt locales](#).

Characteristics and limitations of Form Recognizer

In this section, we'll review what accuracy means for Form Recognizer and how to assess it for your context.

Accuracy

Text is composed of lines and words at the foundational level, and entities such as names, prices, amounts, company names, products at the document understanding level.

Word-Level Accuracy

A popular measure of accuracy for OCR is word error rate (WER), or how many words were incorrectly output in the extracted results. The lower the WER, the higher the accuracy.

WER is defined as:

$$WER = \frac{S + D + I}{N} = \frac{S + D + I}{S + D + C}$$

Where:

Term	Definition	Example
S	Count of incorrect words ("substituted") in the output.	"Velvet" gets extracted as "Veivet" because "l" is detected as "i."
D	Count of missing ("deleted") words in the output.	For the text "Company Name: Microsoft," Microsoft isn't extracted because it's handwritten or hard to read.
I	Count of nonexistent ("inserted") words in the output.	"Department" gets incorrectly segmented into three words as "Dep artm ent." In this case, the result is one deleted word and three inserted words.
C	Count of correctly extracted words in the output.	All words that are correctly extracted.
N	Count of total words in the reference (N=S+D+C) excluding I because those words were missing from the original reference and were incorrectly predicted as present.	Consider an image with the sentence, "Microsoft, headquartered in Redmond, WA announced a new product called Velvet for finance departments." Assume the OCR output is " , headquartered in Redmond, WA announced a new product called Veivet for finance dep artm ents." In this case, S (Velvet) = 1, D (Microsoft) = 1, I (dep artm ents) = 3, C (11), and N = S + D + C = 13. Therefore, WER = (S + D + I) / N = 5 / 13 = 0.38 or 38% (out of 100).

Entity-Level Accuracy

At the document level, for example in case of an invoice or receipt, an error of a just one character in the entire document might be rated insignificant but if that error is in the text that represents the paid amount, the entire invoice or receipt may get flagged as incorrect.

Another useful metric is the entity error rate (EER). It's the percentage of incorrectly extracted entities such as names, prices, amounts, and phone numbers out of the total number of the corresponding entities in one or more documents. For example, for a total of 30 words representing 10 names, two incorrect words out of 30 equals 0.06 (6%) WER but if that results in 2 names out of 10 as incorrect, the Name entity error rate (EER) is 0.20 (20%), way higher than the Word Error rate (WER).

Measuring both WER and EER is a useful exercise to get a full perspective on document understanding accuracy.

Best practices to improve custom model quality

When you're using the Form Recognizer custom model, you provide your own training data so that the model can train to your specific forms and documents. The following list uses the custom form model type to share starter tips for improving your model quality.

- For filled-in forms, use examples that have all of their fields filled in.
- Use forms with real-world values that you expect to see for each field.
- If your form images are of lower quality, use a larger data set (at least 10-15 images, for example).

For a full guide and input requirements, see [Build a training data set for a custom model](#).

Prebuilt model limitations

Form Recognizer prebuilt models are for processing specific document types and are pre-trained on thousands of forms. This allows developers to get started and get results within minutes, with no training data or labeling required. For prebuilts, it's important to note the list of input requirements, supported document types, and locales for each prebuilt model for optimal results. For example, refer to the prebuilt [Invoice input requirements](#).

System limitations and best practices to improve system performance

Consider the following points about limitations and performance:

- The service supports images and documents. For the allowable limits for number of pages, image sizes, paper sizes, and file sizes, see [What is Form Recognizer?](#).
- There are many variables that can affect the accuracy of the OCR results upon which Form Recognizer depends. These include document scan quality, resolution, contrast, light conditions, rotation, and text attributes such as size, color, and density. For example, we recommend that the image be at least 50 x 50 pixels. Refer to the product specifications and test the service on your documents to validate the fit for your situation.
- Be sure to note the limitations of each service with regard to currently supported inputs, languages and locales, and document types. For example, refer to the [Layout supported languages](#).

Customer evaluation

Form Recognizer's performance will vary depending on the real-world solutions that it is implemented for. In order to ensure optimal performance in their scenarios, customers should conduct their own evaluations. The service provides a confidence value in the range between 0 and 1 for each extracted word and key-value mapping. Customers should run a pilot or a proof-of-concept representing their use case to understand the range of confidence values and the extraction quality from Form Recognizer. They can then estimate the confidence value thresholds for the results to be either send for straight-through-processing (STP) or reviewed by a human. For example, the customer may submit results with confidence values greater than or equal to .80 for straight through processing, and apply human review to results with confidence value less than .80.

Learn more about responsible AI

[Microsoft AI principles](#)

[Microsoft responsible AI resources](#)

[Microsoft Azure Learning courses on responsible AI](#)

Learn more about Form Recognizer

[What is Azure Form Recognizer?](#)

Contact us

[Give us feedback on this document](#)

About this document

© 2022 Microsoft Corporation. All rights reserved. This document is provided "as-is" and for informational purposes only. Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it. Some examples are for illustration only and are fictitious. No real association is intended or inferred.

Published: 02/16/2022

Last updated: 02/16/2022