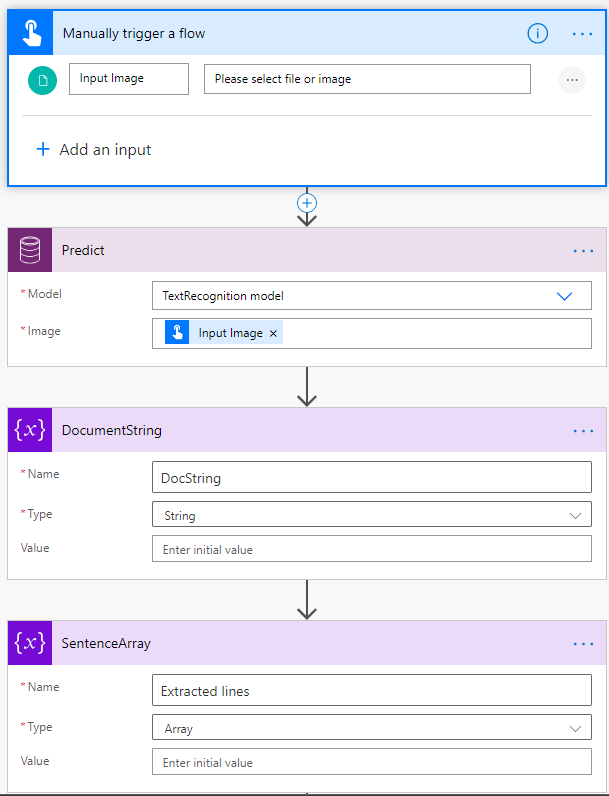
Sign up/Activate AI Builder

## Obtain text from PDF document using text recognition

* 1. Create new flow (Text Recognition)
     1. Manually trigger flow with file input
     2. Add “Predict” action using Text recognition model and input image defined in manual trigger.
     3. Instantiate string variable for text (This variable will store all recognized document text.)
     4. Add apply to each action using “results” from previous steps.
        1. Add apply to each action using “lines” as the input, then use the variable “append to string” action to append all lines.
     5. Output to file or use scope to view variable in flow. (example text below)



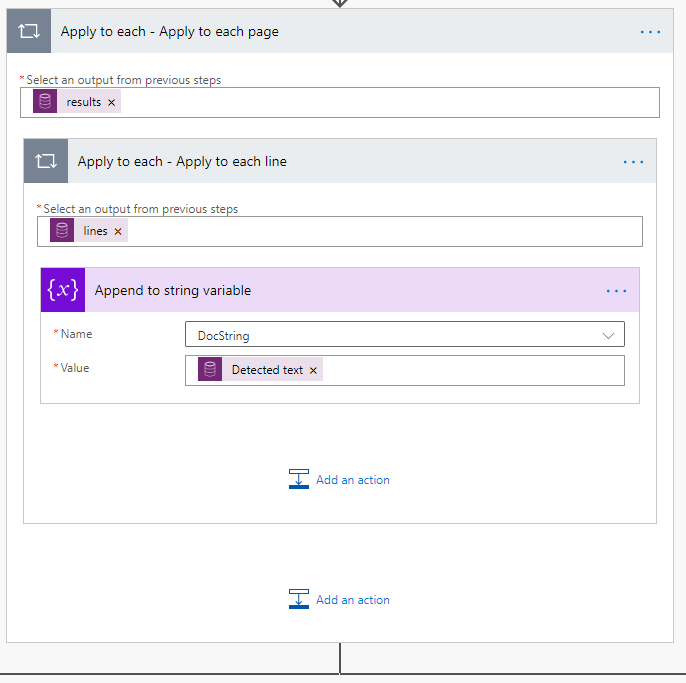
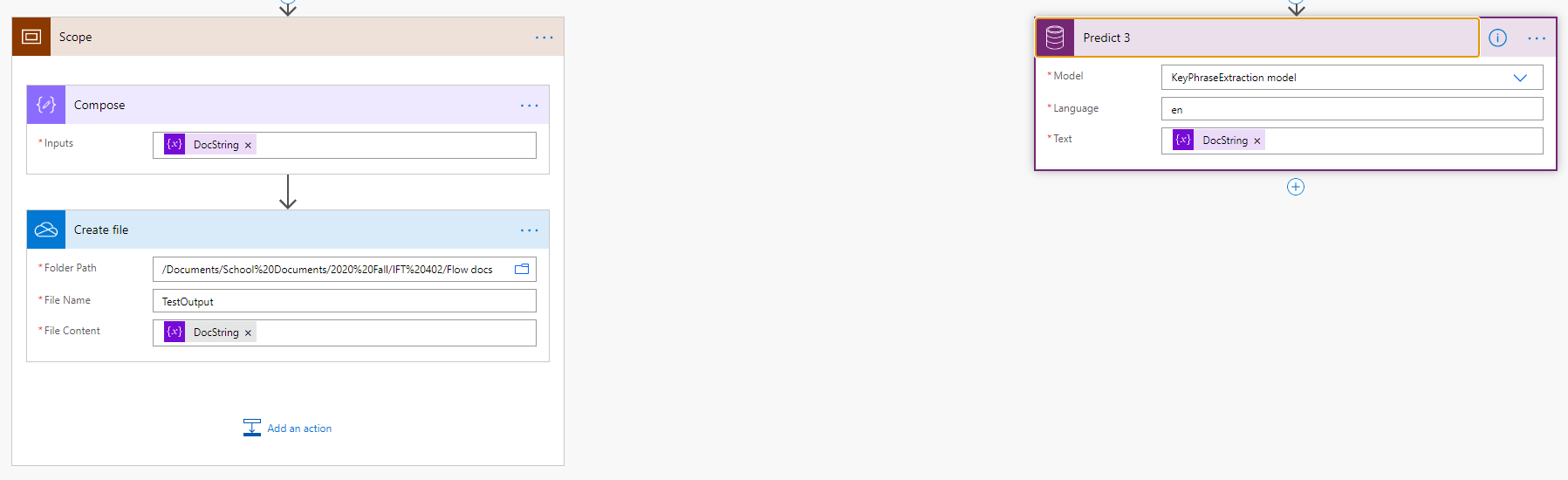
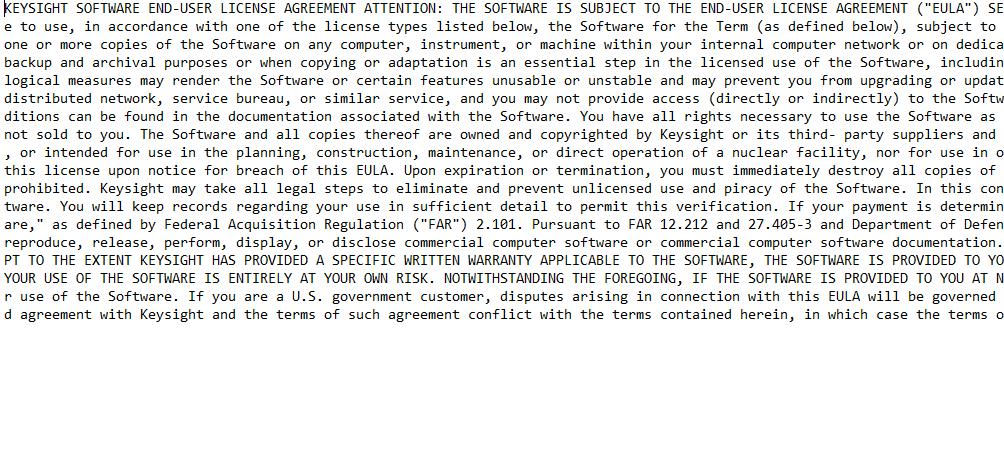


Figure : Space must be added after "Detected text"

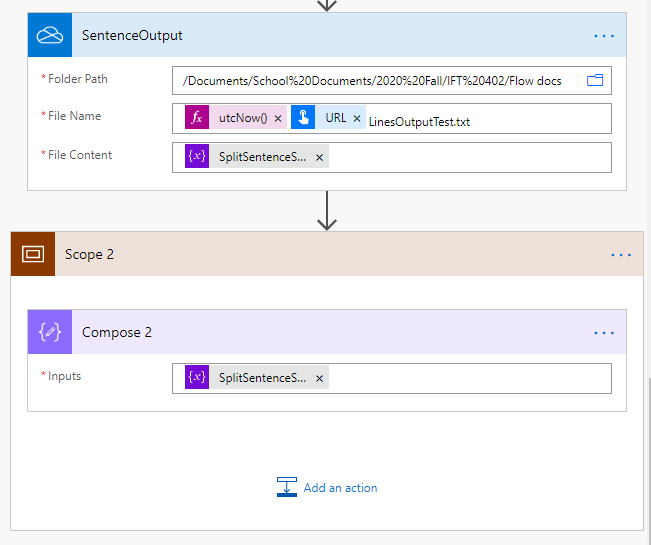
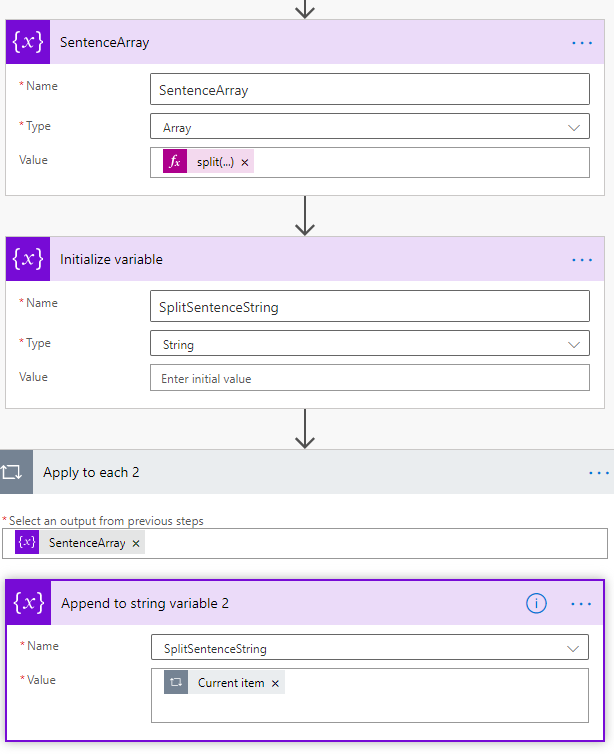


* 1. Output



## Key Phrase Extraction

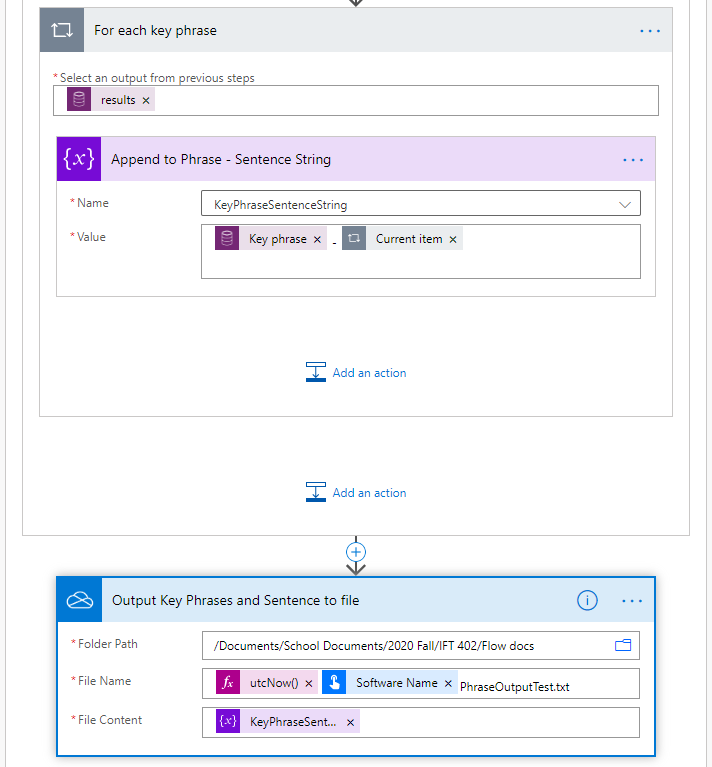
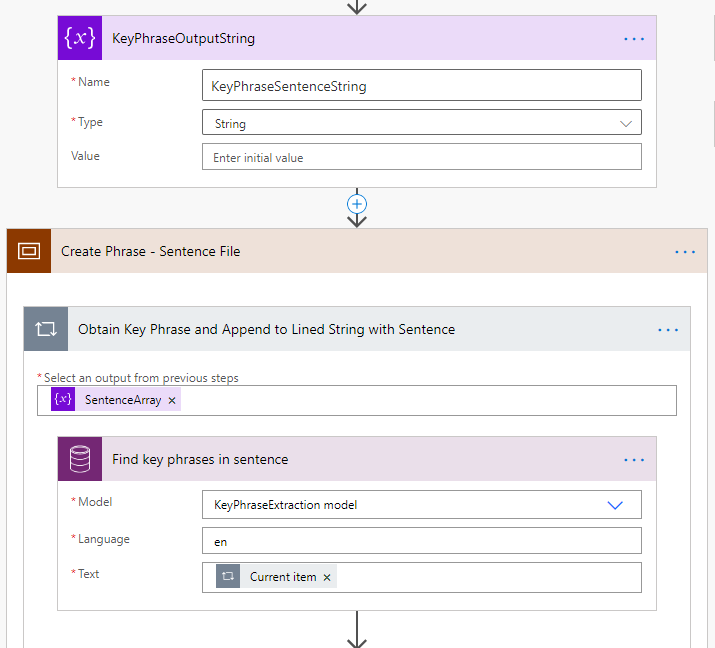
1. Split string into sentences



* + 1. A parallel action was created next to the “DocumentString” action.
    2. The DocString varable is split into SentenceArray by using “split(variables('DocString'),'. ')”
    3. A new variable SplitSentenceString is created to append sentences
    4. Each element is appended to SplitSentenceString with a carriage return
    5. The populated SplitSentenceString is Outputted via OneDrive and scope for viewing in power automate.

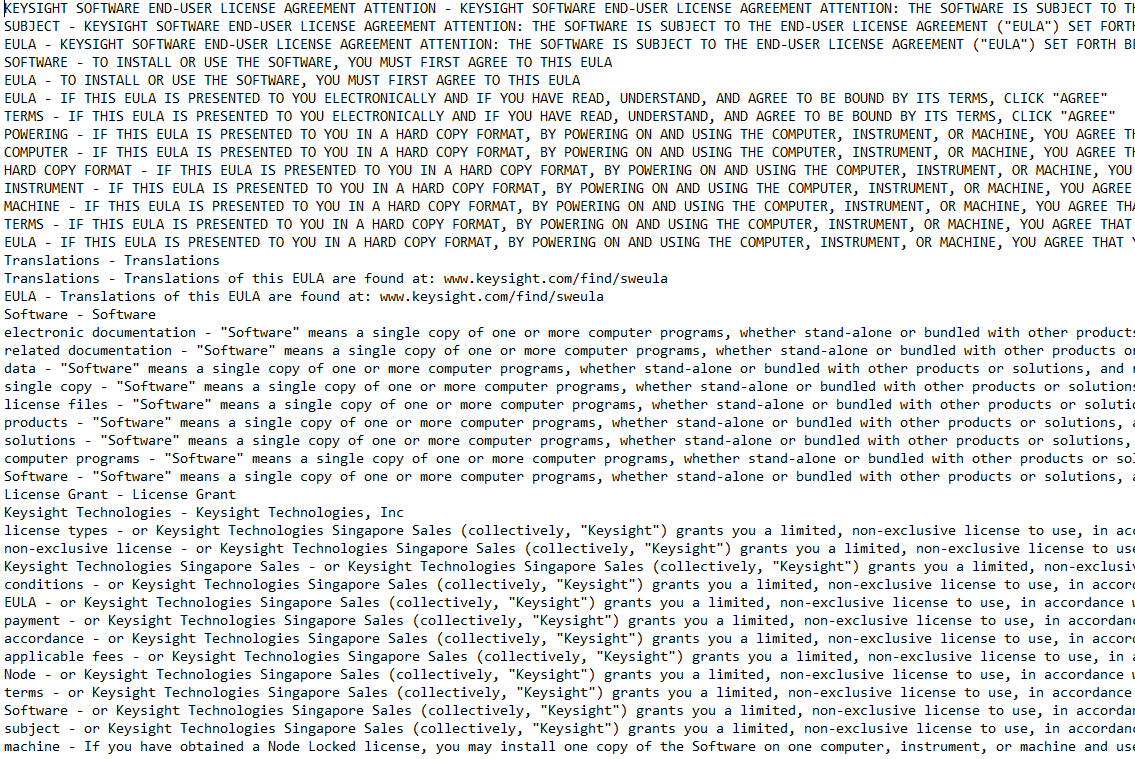
NOTE: Sentences split by delimiter causing inconsistent sentences. No other method has been discovered to tokenize in Microsoft Flow.

1. Add key phrase to define each sentence
   * 1. Apply key phrase recognition
     2. Concat phrase and sentence
     3. Output to file



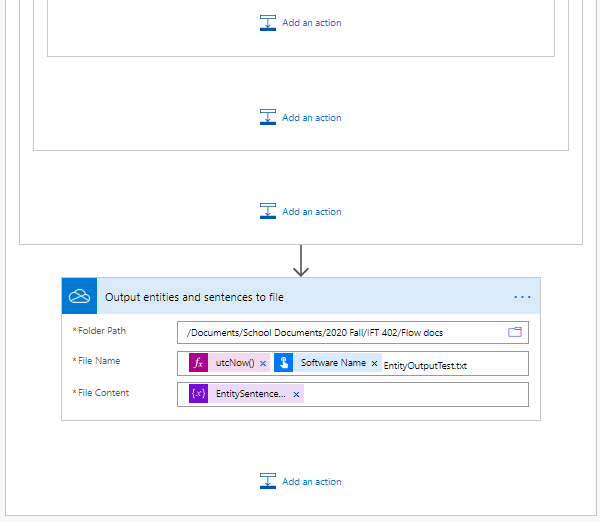
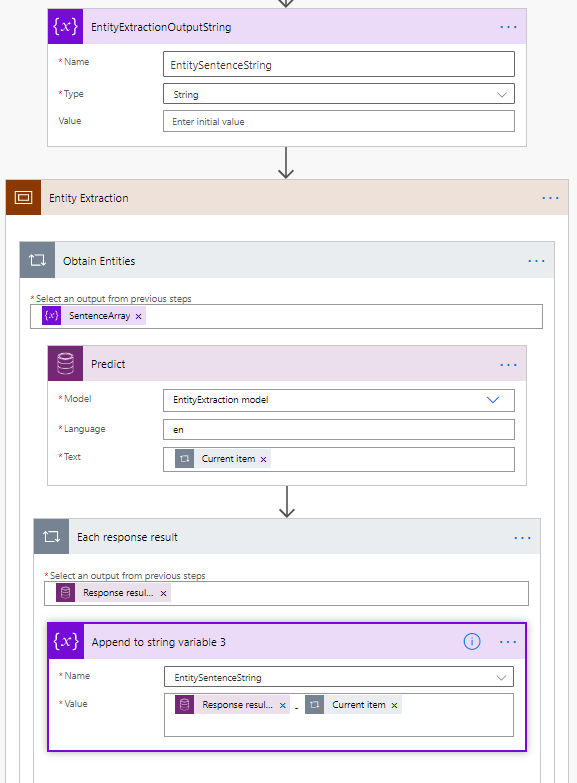
NOTE: Errors occur with strings that exceed 64 characters (e.g. URL) which prevent file processing.

* 1. Example Output

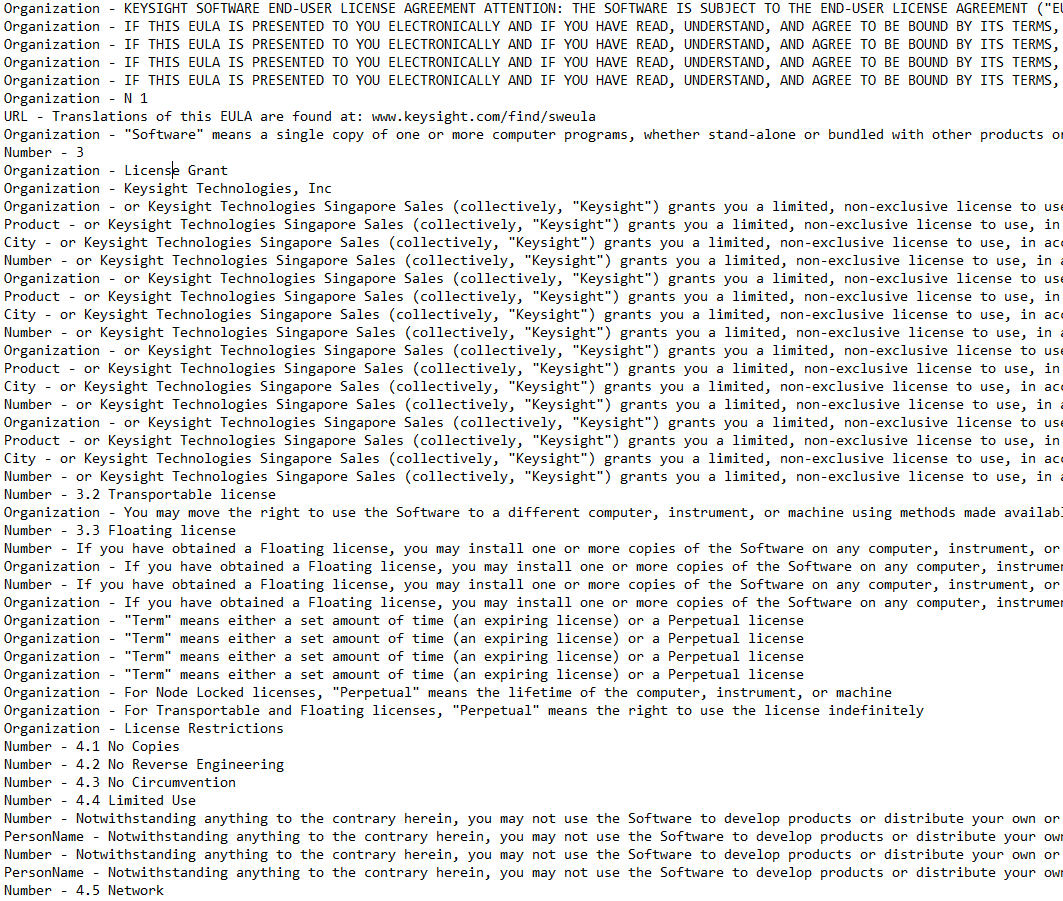


## Entity Extraction

1. Entity extraction is almost identical to the key phrase extraction process
   * 1. Create variable to store entities and sentences
     2. Use scope to contain the processes (optional)
     3. Obtain entities by using the SentenceArray output
     4. Run the prediction model for entity extraction using “current item” in the text field
     5. Obtain the entity type for each response result
     6. Perform append to EntitySentenceString
     7. Output EntitySentenceString to file

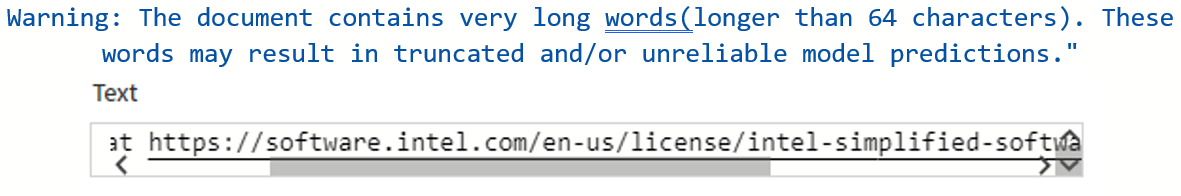


1. Output example

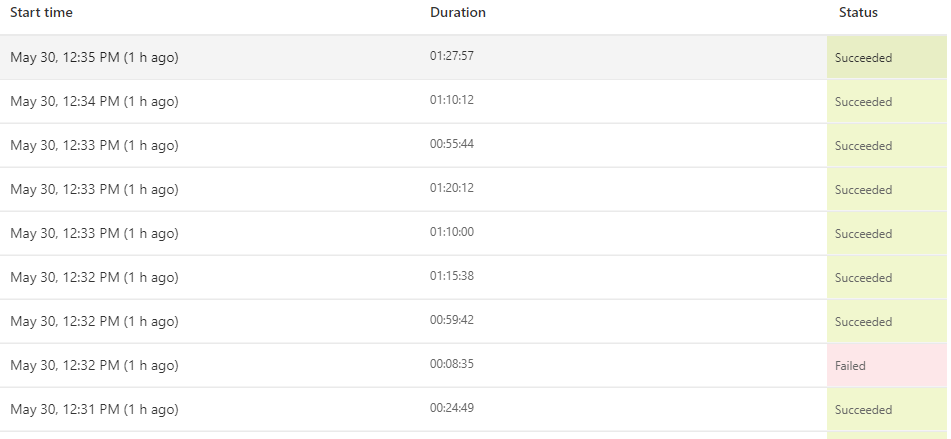


## Performance

Text recognition accurately converts images to text, however, the organization of the text depends on the source document formatting. The output for documents containing text in multiple columns were out of order. Documents containing “recognized” words longer than 64 characters caused a flow failure and would not provide any output for the failed action.



Processing time for the previous operations ranged from 20 to 90 minutes.



## Analyze with category classification

Train the model

Cat class steps