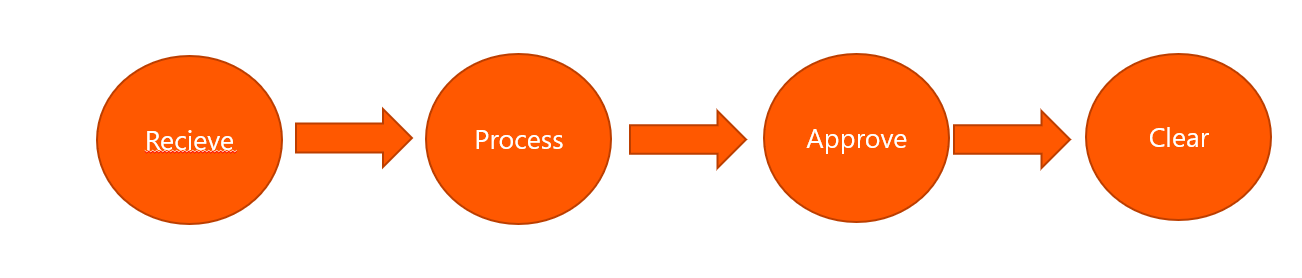
**INTELLIGENT OPERATIONS**

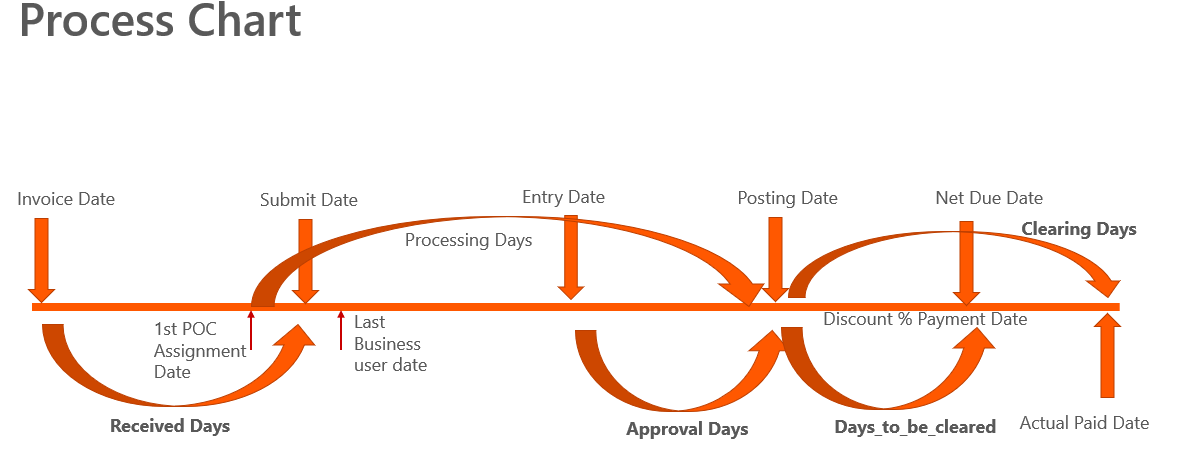
**MS BPO Analytics Project**

In terms of business, an Invoice is a request submitted to Microsoft Operations for processing an purchase order or refund or expense. A typical Order-to-Cash process for businesses involves accounts receivable collection after an invoice is issued to the customer. A usual payment term of 30, 45 or 60 days is often provided for the customer to make full payment of the invoiced amount. However, many businesses face a similar problem of having customers who do not pay on time, and will have to take intervention actions to remind their customers to pay their outstanding invoices. Such intervention actions cost money and time

Invoice Payment predictions can be used to optimize payment strategies. By identifying invoices that are likely to be paid late, operations team can focus on customers who are likely to have large amounts of overdue, which in turn can lead to lower Days Sales Outstanding (DSO) and better visibility into future cash flow. An Invoice undergoes following four main stages:



The above mentioned four stages can still further explained by the below diagram based on the date it has taken to process



By understanding the business and getting input from the operations team. Dates columns are converted mainly into below four days

1.Received Days

2.Processing Days

3.Days to approval

4.Days to be cleared

5.Clearing days

The formula to calculate those days are mentioned below. (If nullvalue in Formula 1 Apply Formula2)

|  |  |  |
| --- | --- | --- |
| **Stages** | **Formula1** | **Formula 2** |
| Received Days | Invoice Date – Receipt date | Invoice Date - Submit date |
| Processing days | First POC assignment date - Submit date | Receipt date - Submit date |
| Days\_to\_approval | Submit date-Posting date | Nil |
| Paid Days | Clearing date- Bline date | Clearing date - Invoice date |
| DaystobeCleared | Posting date - Netdue date | Nil |

**COMPLETE FLOWCHART:**

***Invoice Date***

Invoice

Received/Entered by Agent

Yes

Issue on approval related

*Invoice Date, Received Date,* ***First POC Assignment Date***

No

Invoice

*Invoice Date, Received Date,**First POC Assignment Date,* ***Last Business User Date***

Invoice

Submit to System

*Invoice Date, Received Date,**First POC Assignment Date, Last Business User Date,* ***Submit Date***

*Invoice Date, Received Date,**First POC Assignment Date, Last Business User Date/****Submit Date***

Invoice

Processing for final submission

*Invoice Date, Received Date,**First POC Assignment Date, Last Business User Date/Submit Date,* ***Entry Date***

*Invoice Date, Received Date,**First POC Assignment Date, Last Business User Date, Submit Date,* ***Entry Date***

Invoice

Approved by System

*Invoice Date, Received Date,**First POC Assignment Date, Last Business User Date, Submit Date,**Entry Date,* ***Posting Date***

*Invoice Date, Received Date,**First POC Assignment Date, Last Business User Date/Submit Date,**Entry Date,* ***Posting Date***

Invoice

Discount

Invoice

Invoice

Invoice

Amount Paid to Vendor

Yes

No

*Invoice Date, Received Date,**First POC Assignment Date, Last Business User Date/Submit Date,**Entry Date,**Posting Date,* ***Payment Date***

*Invoice Date, Received Date,**First POC Assignment Date, Last Business User Date, Submit Date,**Entry Date,**Posting Date,* ***Net Due Date***

*Invoice Date, Received Date,**First POC Assignment Date, Last Business User Date/Submit Date,**Entry Date,**Posting Date,**Payment Date,* ***Actual Paid Date***

*Invoice Date, Received Date,**First POC Assignment Date, Last Business User Date, Submit Date,**Entry Date,**Posting Date,**Net Due Date,* ***Actual Paid Days***

***Invoice Date***

Invoice

Received/Entered by Agent

Yes

Issue on approval related

Received Days

Received Days

No

Invoice

Assignment Days

Invoice

Submit to System

Invoice

Days to Submission

Processing for final submission

Invoice

Days to Approval

Approved by System

Invoice

Discount

Invoice

Invoice

Invoice

Amount Paid to Vendor

Clearing Days

Yes

No

Days to be Cleared

Formulas based on conditions(For deeper Understanding):

* Received Days – No. of Days when Invoice was received by agent/submitted to system, from Bill Date
* Assignment Days – No. of Days when Invoice was in request system for clarification
* Days to Submission – No. of Days when Invoice goes for final submission of approval
* Days to Approval – No. of Days when Invoice gets approved
* Days to be Cleared - No. of Days when Invoice has to be cleared once approved
* Clearing Days – No. of Days when Invoice is cleared once approved
* Actual Paid Days = Received Days + Assignment Days + Days to Submission + Days to Approval + Clearing Days

If Assigned for clarification

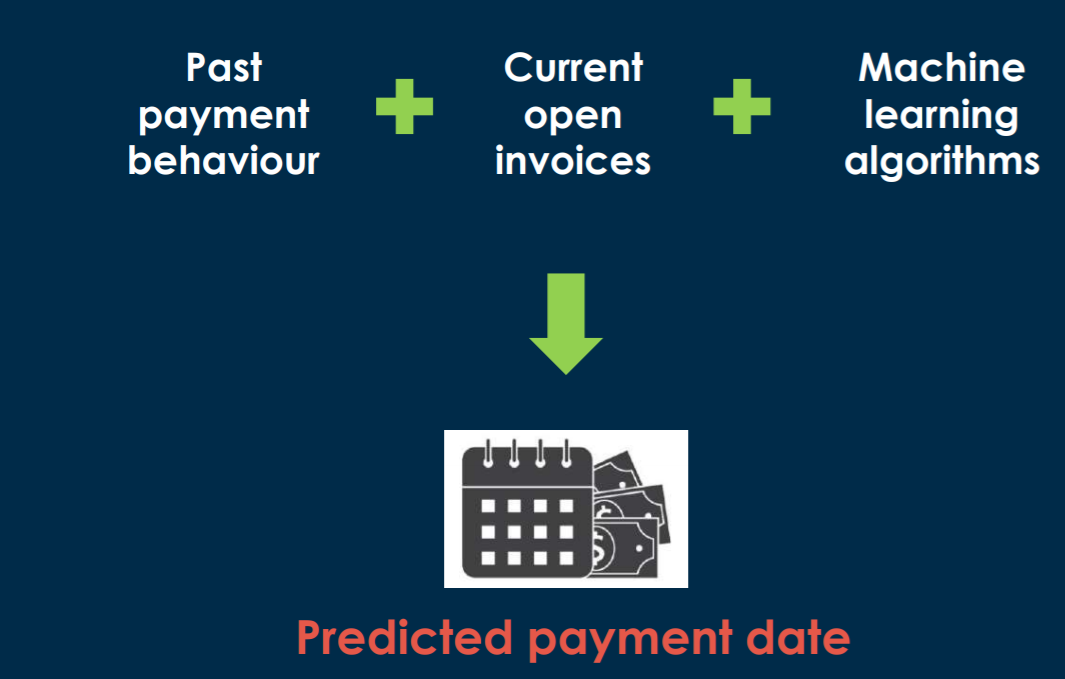
Processing Days = Assignment Days + Days to Submit + Days to Approval + Clearing Days

Payment Term = Processing Days + Days to be Cleared

**MACHINE LEARNING:**

Machine Learning explores the study and construction of algorithms that learn from and make predictions on large volumes of data. It is the science of getting automations to act without being explicitly programmed to do so.

Machine Learning has the ability to process, analyze, and identify patterns amidst the enormous volume of historical data available for each customer. It is able to predict the payment date at an invoice level for all customers and help the collections teams become proactive through improved dunning strategies.



**ALGORITHMS USED:**

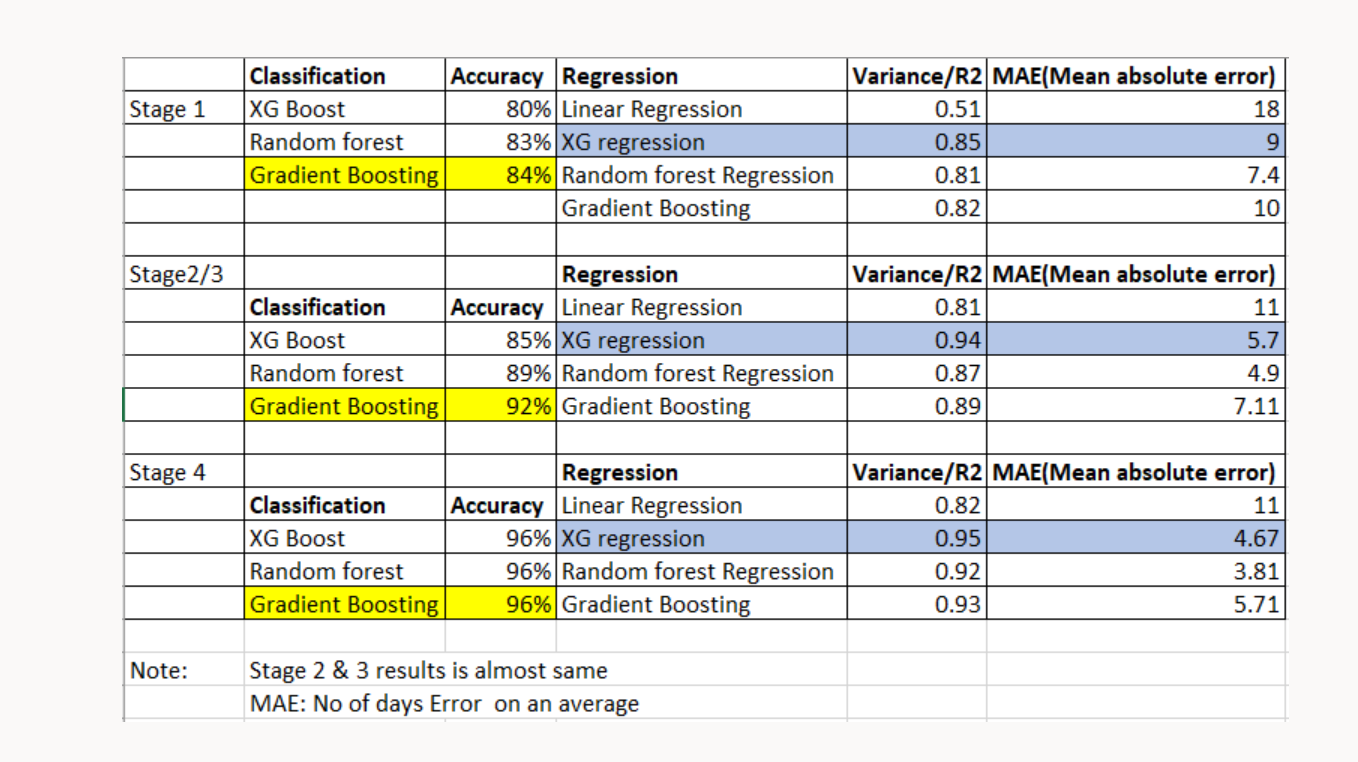
Following are the two algorithms which I have used for the Payment Date Predictions after many experimentations

**Random Forest:** Random forests or random decision forests operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. • Gradient Boost & Adaboost: These are boosting algorithms.

**Ada(Adaptive)boost(Boosting)** is an iterative process that fits a sequence of weak learners on different weighted training data. It starts by predicting an original data set and gives equal weight to each observation. If the prediction is incorrect using the first learner, then it gives higher weight to observations which have been predicted incorrectly. Gradient boosting minimizes the loss of the whole system using the Gradient Descent method.

**INVOICE DATE PREDICTION RESULTS:**

Two types of predictions The predictions are done for all the stages and the results are as below:



**CLASSIFICATION & REGRESSION:**

In the classification model,Invoices are classified based on the payment status into

1.Early Payment

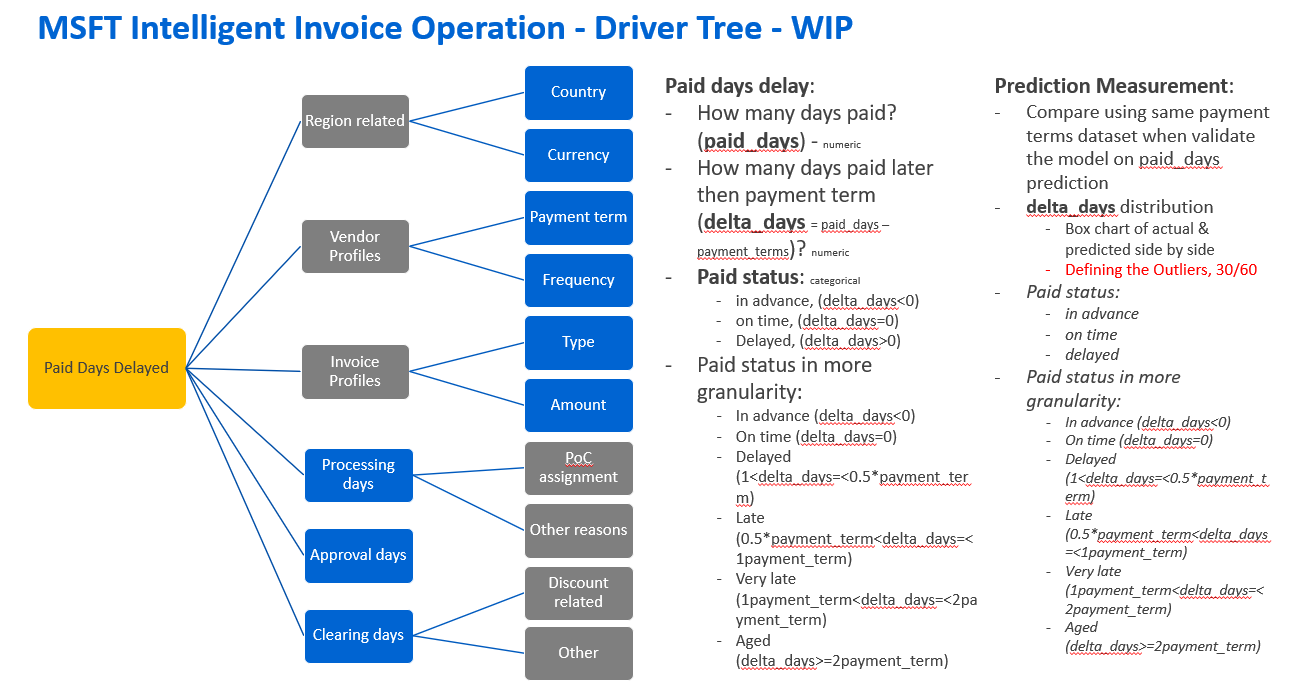
2.On time Payment

3.Delay payment

As the business require the exact date ,we didn’t concentrate much on classification,So we moved on to Regression which gives the exact payment days

**Target :Paid\_days**

Paid days is getting affected by various other features in the input column. This is explained clearly using the driver tree below:



II.Purchase Order (PO) Prediction

The same application is used in the PO Prediction for predicting the purchase order invoice date

AUTOMATION:

The python files is converted to batch file and scheduled everyday in window scheduler in order to get the predictions daily basis

2. **Text Analytics for MS-CRM**

Tickets are generated in daily basis

CURRENT CHALLENGES:

* Too many emails received by the CRM Agents in a single day enquiring about the Invoices/PO’s
* Agents have to open the emails and categorize to identify what the email is about and belongs which Vendor/Invoice it is referring.
* Too much manual effort and time consumed
* Priority emails could be left out

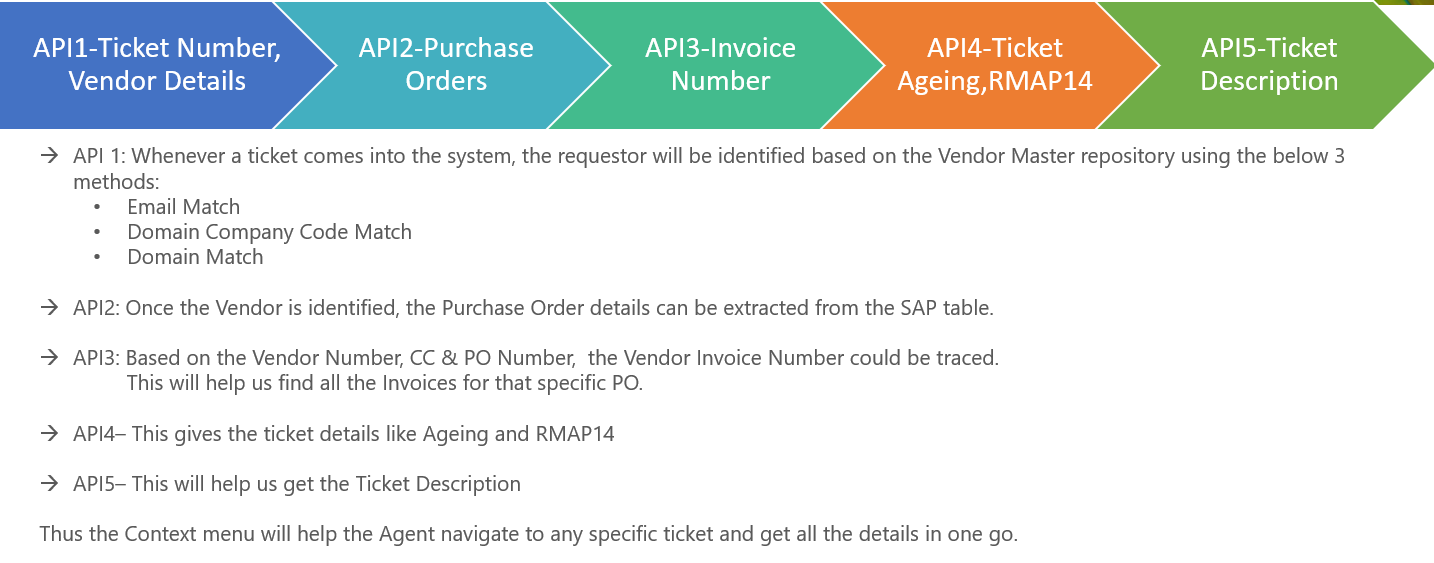
SOLUTION

The Solution has 2 parts:

* + Context Menu built using 5 API’s
  + Every email coming in will be sorted using Text Analytics/Text Categorization.

Thus the Agent can have information about what the email is about and can quickly get the details from the Context menu to validate the PO/Invoice and the Amounts referred to in the email.

CONTEXT MENU:

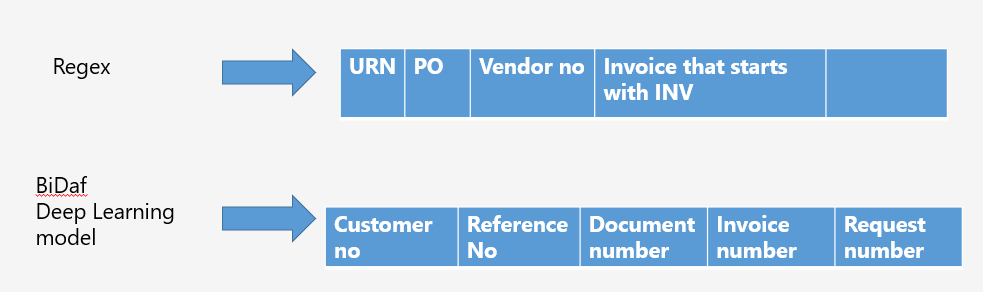


* API 1: Whenever a ticket comes into the system, the requestor will be identified based on the Vendor Master repository using the below 3 methods:
  + - Email Match
    - Domain Company Code Match
    - Domain Match
* API2: Once the Vendor is identified, the Purchase Order details can be extracted from the SAP table.In order to extract the vendor number Regular Expressions are used
* API3: Based on the Vendor Number, CC & PO Number, the Vendor Invoice Number could be traced. But if we need to extract from the mail directly,we don’t have specific format for Invoice number(like starting number , no of digits), So we have used Deep learning model which will be explained in detail later.

This will help us find all the Invoices for that specific PO.

* API4– This gives the ticket details like Ageing and RMAP14
* API5– This will help us get the Ticket Description

Thus the Context menu will help the Agent navigate to any specific ticket and get all the details in one go. The below diagram explains about the models which we used to extract various numbers from the mail.



Deep learning:

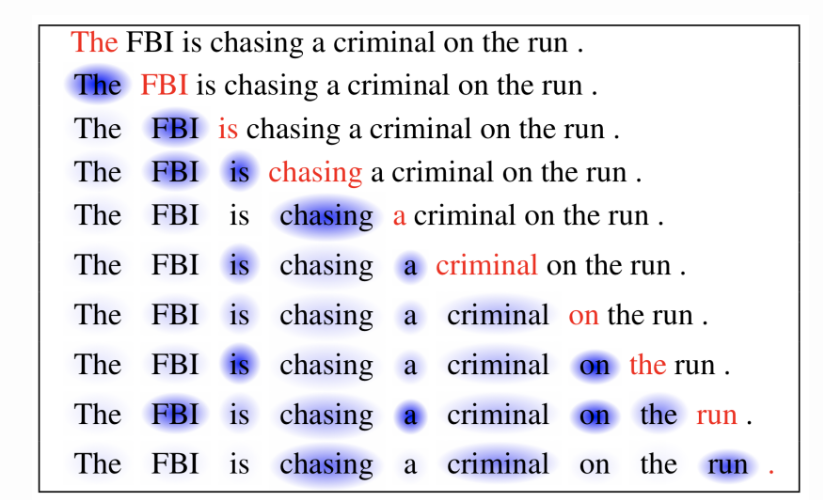
In order to extract the Invoice number and other email, generalized Question answering system was created. Let the email be the comprehensions and questions are asked, so it will try to bring up the answers that are closely related.

Question:” What is the Invoice number?”

Answer: 12334443

DEMO of the model: <https://demo.allennlp.org/reading-comprehension>

Working:

We have used Bi**-Directional Attention Flow for Machine Comprehension(BiDAF)** pretrained model from Allen AI. It understands the grammar 

. The purpose of different layers used in bidaf are briefly described below

* **Character Embedding Layer** maps each word to a vector space using character-level CNNs.
* **Word Embedding Layer** maps each word to a vector space using a pre-trained word embedding model.
* **Contextual Embedding Layer** utilizes contextual cues from surrounding words to refine the embedding of the words. These first three layers are applied to both the query and context.
* **Attention Flow Layer** couples the query and context vector and produces a set of query aware feature vectors for each word in the context.
* **Modeling Layer** employs a Recurrent Neural Network to scan the context.
* **Output Layer** provides an answer to the query

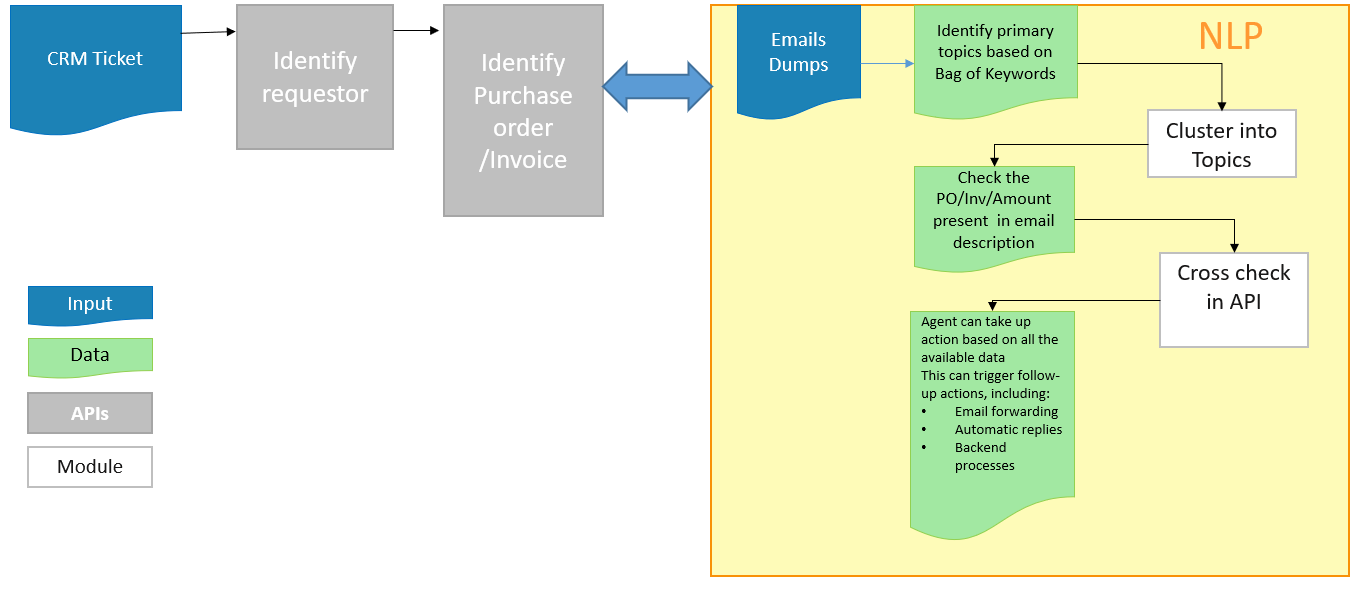
For additional details on the network architecture please refer to the original paper.

<https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1174/reports/2760988.pdf>

TEXT CATEGORISATION

* Using Un-Supervised NLP models, Text Analytics/Text Categorization will be done for the Emails coming into the system.
* Based on the rules, it will be clustered into certain Primary and Secondary topics using these AI/ML models.
* This will help us split the email on what is the type of Query
* E.g. If an email has a keyword like ‘Invoice/Inv’, it could be from a Primary Topic called ‘Invoice Management’.
* This will help us understand that the email is regarding a query on certain invoice.
* The next action would be to extract the keywords like the actual Invoice Number, Amount etc.
* So this will give the Agents first-hand information on what the email is about and what is the action required.
* Also the Agent can then go to the Context Menu to verify and cross check on the Purchase Orders, Invoices and Amount the email talks about and reply accordingly.

SOLUTION ARCHITECTURE



**K-means CLUSTERING**: K-Means clustering is a type of unsupervised learning, which is used when you have unlabeled data (i.e., data without defined categories or groups). The goal of this algorithm is to find groups in the data, with the number of groups represented by the variable K. Here in our case ,similar Emails are clustered into single group based on the similarity of words.

**Topic Modelling:** In machine learning and natural language processing, a topic model is a type of statistical model for discovering the abstract "topics" that occur in a group of emails

AI Components in MSFT :

We have created business content that demonstrates Where Artificial Intelligence and Data Science could be implemented to make the operations effectively by reducing the manual power using Driver tree has explained overall challenges and the use case scenarios below:

