Capstone Project Proposal

e-Commerce Chatbot

Scott Smith

Business Goal

One of the critical components of any eCommerce site's customer experience is the quality of Customer Service. It has a direct impact on repeat business, sales growth, and viral marketing. Net Promoter studies have shown that recommendations from friends or colleagues (what Net Promoter measures[1]), have the most impact of generating new business. What Net Promoter also indicates is that a Customer Service interaction is a moment-of-truth when customer loyalty is made or broken.

Customer Service is also a cost center for an eCommerce business. Using live call center agents cost on average of $15 a ticket[2]. - So there is always pressure to control costs and look for new ways to deliver a great customer experience. The latest tool in the arsenal for service organizations is Machine Learning and Chatbots.

According to IBM Watson, virtual agents (chatbots), can reduce this $15 cost per ticket down to $1[3]. Overall, as stated by Chatbots Magazine, chatbots can reduce customer service costs by up to 30%[4]. There is clearly a cost-savings argument for Chatbots.

This project proposal is to create an eCommerce Chatbot to supplement live chat, email, and call center support.

For most consumers, talking with a person to get a problem resolved is usually the easiest form of interaction, but it also takes the longest time to get someone on the phone and then get a result. And while consumers want to talk with someone, most consumers (and millennials) - use an 'Internet' first approach - can I solve the problem by going to the internet. Thus, providing a natural way to communicate as well as the fast turnaround is the sweet spot for chatbot success.

---

[1] https://blog.hubspot.com/service/what-is-nps
[2] https://www.askspoke.com/blog/it/it-help-desk-statistics/
[3] https://www.ibm.com/blogs/watson/2017/10/how-chatbots-reduce-customer-service-costs-by-30-percent/
[4] https://chatbotsmagazine.com/how-with-the-help-of-chatbots-customer-service-costs-could-be-reduced-up-to-30-b9266a369945

Customer Service conducts a lot of transactions with customers. Many of these transactions are routine. *This proposal is to replace these routine transactions with a Customer Service Chatbot*. Services that can be replaced include, Account inquiries, help with product selection, cross-sell and up-sell, monitoring and replying on social media.

The business benefits are many. Besides lowering support costs, chatbots can: Increase revenue via cross-selling and upsell, increase customer satisfaction, reduce customer churn, reduce new customer acquisition costs, and increase the overall customer lifetime value. In addition, chatbots can free up support agents to focus on higher-value customer interactions.

Business Case

Customer satisfaction has a direct correlation with customer lifetime value and churn. According to the analyst firm Forrester, "*it costs 5 TIMES MORE to acquire new customers than it does to keep the existing ones and it will cost you 16 times more to bring a new customer up to the same level as the current one.*"[5]. From a Harvard Business Review Article, " *increasing customer retention rates by 5% increases profits by 25% to 95%*"[6].

In our go-to-market plan is roll this out incrementally by replacing routine transactions with chatbots. These are the current statistics for our analysis:

- Current costs: $15 per support call, 1,000,000 calls per year totaling $15 million annually.
- Support growth at 100,000 new support requests year.
- 80% of transactions are routine account transactions.
- Bot costs are estimated to be $1 per transaction (via industry metrics).
- Proposed to replace 10% the first year, increasing over time.

Based upon the industry numbers, this pencils out as follows:

	Current	**Year 1**	**Year 2**	**Year 3**	**Year 4**	**Year 5**
Totals Transactions	1,000,000	1,100,000	1,200,000	1,300,000	1,400,000	1,500,000
Agent Transactions	1,000,000	1,012,000	1,008,000	884,000	728,000	540,000
Cost Per transaction	$15	$15	$15	$15	$15	$15
Total Agent Costs	$15,000,000	$15,180,000	$15,120,000	$13,260,000	$10,920,000	$8,100,000

[5] https://www.superoffice.com/blog/reduce-customer-churn/
[6] https://hbswk.hbs.edu/archive/the-economics-of-e-loyalty

Routine Transactions	800,000	880,000	960,000	1,040,000	1,120,000	1,200,000
Proposed Bot Targets	0%	10%	20%	40%	60%	80%
Bot Transactions	0	88,000	192,000	416,000	672,000	960,000
Cost Per Bot Trans	$1	$1	$1	$1	$1	$1
Total Bot Costs	$0	$88,000	$192,000	$416,000	$672,000	$960,000
Total Support Costs	$15,000,000	$15,268,000	$15,312,000	$13,676,000	$11,592,000	$9,060,000

Overall, 5-year cost savings are projected to be 33% or $32.4 Million.

5 Year Cost No Bots	$97,500,000
5 Year Costs with Bots	$64,908,000
5 Year Savings	$32,592,000
5 Years Saving Rate	33%

As stated above, there are many additional business benefits, cost savings, and revenue growth, including:

1. Reducing customer churn
2. Provide 24/7/365 support
3. Create a smoother customer journey
4. Replace outdated IVR systems
5. Personalize the customer journey

Application of ML/AI

The goal is to create a goal-oriented, conversational chatbot that will manage routine account management transactions for an eCommerce site.

Goal-oriented chatbots like Siri or Alexa are described in the industry blog as "*Interacting with the machine via natural language is one of the requirements for general artificial intelligence. This field of AI is called dialogue systems, spoken dialogue systems, or chatbots. The machine needs to provide you with an informative answer,* *maintain the context of the dialogue, and be indistinguishable from the human (ideally)*".[7]

We will use chatbots to understand the problem and connect the customer with the solution. This will allow for communication with the customer in a natural format and solve the service request quicker rather than having someone wait on the phone.

Initially, we will use chatbots to:

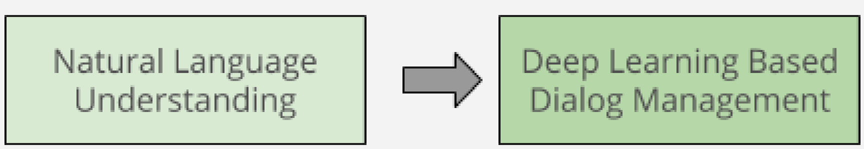
- Understand the customer request. We will focus on account management for our MVP. Tasks include:
    - Reset password
    - inquiry on order status
    - report a defective product
    - process returns
    - pay a bill
    - ...
- The chatbot AI/ML model will process, and categorize the text message.
- Ask for clarification, if needed
- Connect with the back-end systems and provide an answer

Business objectives we are seeking include:

- Take the human out of routine customer interactions
- Allow for customer service reps to provide higher-value services
- Respond to customers faster
- Create higher customer loyalty and customer satisfaction

ML/AI Method

The science of Machine Learning provides multiple ways to process chatbots. The machine learning method that we use will be a retrieval-based model that uses a Machine Learning Classifier to understand the request and map it to the appropriate label class. Training is done via message pairs (the message request and response).



[7] https://blog.statsbot.co/chatbots-machine-learning-e83698b1a91e

The basic flow is to use NLU to understand the input of unstructured text and the deep learning engine to generate responses. We will use a pre-build NLU and add domain-specific ontologies.

We will use a TensorFlow [DNNClassifier] for text classification. (DNN Stands for Deep Neural Network). We can use transfer learning and use pre-trained models to jump-start training. For example, TensorFlow-Hub provides [nnlm-en-dim128] - This is a pre-trained text embedding module.

Our training data can be defined as a closed domain. That is, we have a limited set of inputs and outputs, and we can control the training data accordingly. While we will generally have balanced data, we require multi-class classification to map to the different intent types we need to process.

Program Success Metrics

This program's goal is to improve the support customer experience and reduce costs. To measure program success, we will use existing measures and gauge changes over time. There are three areas we can measure: chatbot, customer, and support metrics.

Chatbot metrics can be used to measure adoption. Using rates of chatbot usage as well as live chat statistics. We can measure:

- Number of active users using chatbots. Adoption success is measured by increases in this number.
- Number chatbot sessions. Adoption success is measured by increases in this number.
- Average chats handled by bots vs. live chat. Adoption success is measured by increases in chatbot usage with decreases in Live Chat
- User Engagement: How well the bot engages - can the bot handle the request, or does the request need to be sent to a live agent?
- The success rate of the bot (requests not assigned to call center)
- Text analytics feedback - direct feedback from the user
- Reuse rate - how often a customer returns to use a chatbot. This should increase over time.

Customer Metrics

- Customer Churn
  - Measure customer activity to calculate churn rates
  - Success is measured in decreased customer churn
- Increase Customer Satisfaction and Net Promoter scores

- Key Drivers for Net Promoter Score (NPS). These are events that impact the NPS score. Poor support should be less of a key driver.

The Customer Support organization is metrics-driven, so there are many metrics we can use to measure chatbot adoption and success. These include[8]:

- Lower per-case support costs. Chatbots should be replacing call center agent.
- Improved support time-to-resolution. Chatbots will have lower resolution times lowering overall time-to-resolution.
- Overall Ticket Backlog. Should decrease over time
- Overall Resolution Rate. Should increase over time
- Average time to reply. Should decrease over time.
- Overall Average first response time. Should decrease over time.
- Overall Average handle time. Should decrease over time.

Data

The training dataset is the essential aspect of training a chatbot. For conversational, goal-oriented chatbots, we need a large number of conversation logs. These logs are message pairs that represent a question or statement from the customer and the appropriate reply.

The chatbot will be trained to understand the customer statement and provide an appropriate reply. When the response is an action, such as providing order status, the chatbot will connect with the appropriate back-office system to return a result.

One key aspect of the training is that this will not be an open-ended chatbot. As a goal-oriented chatbot, the 'conversation' is designed to lead to a conclusion - some action performed or information requested by the customer. This makes the training task more manageable and also limits the chances for bias.

Data Acquisition

Data Sources

The core of the chatbot training is message pairs. These are questions/statements and responses. This data can be obtained from multiple sources, both commercial and open source. For our purposes, we will be operating inside of the constraints of what the organization already performs. For this, we can obtain data from:

- Support tickets
- emails threads

[8] https://www.groovehq.com/support/customer-service-metrics-template

- Live chat transcripts
- Call Center transcripts
- FAQ pages
- User Manuals
- Knowledge bases

Preparing the Data and Costs

We will need to prepare the data to get it ready for training. The prep is based upon what actions, or intents, we have to plan for. The goal is to provide at least 10-20 question-answer pairs per intent. More examples are better, but we need to start with a minimum. For example, if we are looking for order status, we might have the following questions for the 'Order Status' intent:

Question	Response
"What is the status of my order?" 'When will my order arrive?" "Has my order been shipped yet?" "What is my order tracking number?"	Your Order: Shipped on July 23 Tracking number X12312312 Schedule to arrive on July 25

The main cost of data will be taking the internal sources of data and use a service like Figure Eight to put the data into the proper form.
https://success.figure-eight.com/hc/en-us/articles/202703165-Job-Costs-FAQ#cost_estimated

For our purposes, we have the following intents we need to train for

Account Questions / Actions (12 account actions)

- Change Password
- Lost Username
- Change email
- Report lost package
- Find a missing item
- Incomplete shipments
- Order Status
- Account Balance
- Damaged Item
- Update payment information
- Change of Address
- Change of contact information

User Manuals for products. For chatbot How-to questions

- 5,000 products manuals (20 basic questions. I.e. “How do I install X”, “How to troubleshoot Y”, where X & Y would be one of the 5000 products)
- Knowledgebase (500 articles, 20 basic questions)
- How-to (200 articles, 20 basic questions)

FAQ (20 Items)

- Return Policies
- Shipping rules
- Damaged goods policy

The elements we need to train/plan for include:

- Chatbot conversations
- Intents (action to be taken, like lookup order status)
- Intent responses / conversations

Example:

Player	Bot	Backend System Intent
Customer	What is my order status?	
Intent		Order Status Lookup
Conversational Response for Intent	I can help, can you give me an order number?	
Customer	Yes, Order is A123	
Intent		Looks up order
Bot	Your order will be delivered July 25	

This will result in roughly 100 intents (actions) with 10 examples per questions plus conversational aspects of the intents (2-10 per intent with 10 variants for each conversational aspect), yields the potential of 40,000 to 100,000 rows of data (more likely around 40,000 since not all intents will have 10 follow-up questions). This will translate into 10,000 pages to annotate. We will have multiple people reviewing (judgments).

Total Question Pairs = 100 Intents * 10 examples per intent
* 10 conversational elements * 10 examples per element
= 100,000

If we used a service like Figure Eight to annotate the data, we could estimate cost with their formula[9]:

Formula:

(Judgments per row * (Pages of work * Price per page)) + transaction fee + buffer = estimated job cost

Using the above equation, a job with 100 potential questions, with as a minimum of 10 question variants will yield:

- Rows per Page set to 10
    - Note: 1 of these 10 rows will be a test question on every page of work
    - 100,000 rows/ 10 yields about 10,000 pages of work for the job
- Judgments per Row is set to 3
- Price Per Page is set to 12 cents The Basic cost estimate is: **3 Judgments per row * (10,000 Pages * .25 Price per page) = $7,500**

Additional costs will include staff to collect and organize the data for training and define test questions. This will be the bulk of the costs that are estimated at around 4 man-months at $10,000 per month.

Overall, the data prep should cost between $50,000 and $75,000

Personally identifying information (PII) and Security

Most questions will not require PII or security. These include how-to, manuals, and FAQ questions. Account-level questions will be conducted behind a password-protected firewall. The exceptions will be password change requests and order status with a known order number.

In no case will the chatbot request or accept PII data. Anytime that is required, an account-protected form will be provided (such as change of address requests.)

[9] https://success.figure-eight.com/hc/en-us/articles/202703165-Job-Costs-FAQ#cost_estimated

Ongoing sources of data

Ongoing sources of data will include additional live chat logs, new KB articles, new product documentation, and chatbot logs (corrected based upon customer feedback). Each of these will need to be organized and added to the training set. It is expected that batch updates would happen on a 1 to 3-month schedule.

Data Source

The initial size of the training data set to train the first chatbot consist of, at the high-end of the estimate, about 100,000 rows of data. - it is estimated that a minimum of 100 question types, representing about 100 intents (actions to be performed), with approximately 1,000 total question-response pairs plus conversational aspects (10 per question type with 10 variants each) will get us to about 100,000 rows of data.

The questions will be mined from existing corporate data. To this end, there will be bias in the data that might limit the scope of a deployment.

These include:

- Regional English. The initial training will be on US English, for example, local dialects (boot vs. trunk in the UK for example) or slang might not be accounted for.
- Limited to specific intents - The chatbot will have a basic vocabulary and will not be able to manage a general conversation ("How is the weather today").

The data set will consist of:

- Question Pairs (question and response)
- Response - Intent Mapping (taking a response and mapping it to an intent, like order lookup. The response will then use the appropriate intent)
- User Manual Mappings (map the user manual intent to user manual web locations).
- Knowledge Base Mappings (map the knowledge base intent to articles web locations).

Improving the data

During actual use, there will be multiple approaches to measuring accuracy and improve the data. These approaches will include:

- Active monitoring, with a person in-the-loop to provide realtime feedback.
- Survey feedback from customers (after the chat completes)
    - Review and evaluate what should have been replied

Corrections will be made to the source data or additional question pairs, and conversational elements will be added.

Choice of Data Labels

The purpose of the e-commerce chatbot is to respond to customer questions around account management task, product questions, or ordering questions.

The basic training for a chatbot is done with message response pairs, with the pairs in sentence form. Additionally, we need to map different questions to intents (actions to be performed). In addition, the Intent may have follow-up questions that will be used for clarification and for collecting required data for intents.

We picked this scheme since message pairs are the most common method for training chatbots. Message pairs can be converted easily in vectors to be fed into the DNN Classification Model. This is also the way data needs to be feed into the Deep Neural Network Classifier.

The strength if this labeling approach is its simplicity. The question-response pairs, with the response being the classification class, is easy to manage. Responses classes are simply mapped to intents. Our goal is to provide a good user experience. This begins with the training data. Being able to structure our data in this simple format allows for better quality control and the use of 3erd parties to help prepare the data.

Approach Strengths and Weaknesses

Strengths	Weaknesses
Simple text. This will make it easy to create message pairs to manage classification.	May limit a more complex understanding of the text.
Simple classification of the response text. Response text will map directly to intents. Thus, there is no need to train for specific intents.	May cause problems of responses work for multiple intents.
Followup questions do not require specific labeling or separate training.	Not true conversation, but rather, followup questions are used to collect additional data for intents.

We do not need specific mappings or labels for follow-up questions. We only need to map specific question pairs and then map question answers to intents. (This may not need to be trained.)

Question Pairs

Question	Response
What is my order status?	I can help, can you give me an order number?
Can you tell me when order A345 will be delivered	Here is your order status
My Order number is A123?	Here is your order status
How do you install XYZ?	Here is the manual for XYZ
I need help changing my password	You can change your order here.

Intent Mappings

Response	Intent
I can help, can you give me an order number?	Order Lookup
Here is your order status	Order Lookup (Followup)
Here is your order status	Order Lookup
Here is the manual for XYZ	User Manual Lookup
You can change your order here.	Password Update

This works to a general mapping of requests to actions:

- Customer asks a question
- Topic area identified (Intent)
- Bot responds, requests addition info based upon intent
- Customer responds
- Bot performs the action

Model

Building a chatbot requires what functions we want the chatbot to perform. Since we want the chatbot to supplement live chat and call center agents, we want the chatbot to be more human-like. That is, it understands natural language and understands context. While the initial plan is to create a chatbot that is goal-oriented, replacing common support functions like password resets, we also want the chatbot to grow into a sales and recommendation engine.

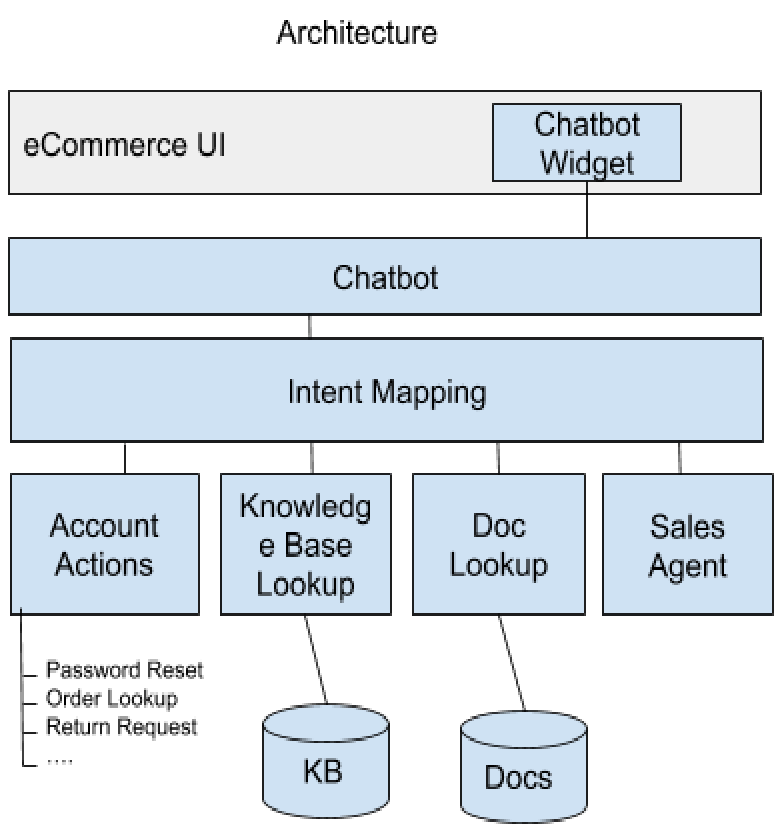
Model Building

Goal-oriented, conversational chatbot - solve specific user requests. There are several approaches that we can use to build; they include:

- Using a commercial development vendor
- Use services
- Use open-source solutions
- Build the model in-house

Besides building and training a model, we also have development efforts to add a chatbot to the eCommerce platform and mobile devices, but also integrate with back-end systems. Reviewing a sample architecture, we can see that the chatbot is only one part of the solution. There will be in-house development to perform these integrations.

There are pluses and minuses to how the implementation should proceed. Looking at the two choices, we can see that we would still need to do 5 out of 6 tasks in-house. We will have security issues if we use an outside source.



Tasks

	In-House	Services
Bot Model Development & training	Requires in-house expertise	Expertise outsourced
Bot Data	In-house data prep	In-house data prep
Integration	Requires in-house developers	Requires in-house developers
Support Use Case	Yes	Yes
Security	We control	Out of our control
Access to Data and backend systems	in-house, secure	Potential security risk

Given that a large part of the costs will be integration and data prep, it makes sense to keep in-house based upon cost and the wide availability of open-source chatbot frameworks. 20% of the cost will be Chatbot development.

Estimates that bot development will take 120-160 hours and cost upwards of $30,000 for commercial systems[10]; this only is part of the cost since we need to consider integration and data prep as well. This is only for one chatbot. Our goal is a chatbot that will manage multiple customer interactions (Account, Information, Sales). Thus the cost will be higher (up to $100,000) since we may need to create additional chatbot models (such as sales support and product recommendations.)

In house development, with the hiring of a Data Scientist to help, would be the best way to go. Even if we went to a 3rd party, we still need in-house expertise to manage relationships and understand the on-going needs of the chatbot. Using open source solutions[11] to aid in the actual building and training the model will also give us many of the benefits of going to a 3rd party, but without the cost.

One such free solution is Rasa (rasa.com). The main benefit of Rasa is that it starts with a free model, but as our needs grow it as more advanced enterprise features in a paid model. Rasa includes Natural Language Understand (NLU) for understanding user messages and a backend for managing conversations. Rasa uses Tensorflow for machine learning, which is very popular and well documented for building chatbots.

[10] https://appinventiv.com/blog/how-much-is-chatbot-development-cost/
[11] https://blog.verloop.io/the-best-open-source-chatbot-platforms-in-2019/

Evaluating Results

The chatbot will use a Deep Neural Network Classifier for training.

The primary metric for evaluating the performance of the model during training is **accuracy**. That is, the DNNClassifier uses the accuracy metric to evaluate each training epoch (a complete run through the entire data set). Training will consist of multiple epochs, and we can use the accuracy score to determine what epoch is the best.

In training the model, we will also monitor the epochs and plot the training and validation loss. Monitoring would be to prevent over or underfitting.

Classification models have other metrics to measure training results and how well it will predict live. We can use these metrics and adjust our training for our specific needs. These can be explained through the metrics: Precision, Recall, and F1 Scores.

Precision

Precision is defined as how many of the actual class of messages did we predict correctly. Low Precision means we are classifying chat messages incorrectly. For example, a customer asked for order status, and we think they asked for a password change. High Precision reflects an accurate response and better customer experience.

Recall

Recall is defined as a percentage of how many messages we predicted were correct. A low Recall means we are missing requests for a specific class, for example, order status.

> **From Wikipedia:** Precision can be seen as a measure of exactness or quality, whereas Recall is a measure of completeness or quantity. ... In simple terms, high Precision means that an algorithm returned substantially more relevant results than irrelevant ones, while high Recall means that an algorithm returned most of the relevant results.

The choice of measure, Recall, or Precision, can be difficult. Ideally, we would get everything correct, but for our purposes, we want to maximize correctness with the type of user experience we want to create. This means we want high Precision because we don't what to predict incorrectly and high Recall since we do not want to miss specific requests like order status.

Since we want to maximize both evenly, we can use the F1 Score. This score takes both Precision and Recall into account.

F1
The F1 Score is a metric that uses both Precision and Recall to measure accuracy. F1 is used to get an overall, general score. In more mathematical terms the F1 score is the weighted average of the precision and recall scores. A high F1 score means that we are predicting messages correctly in most of the chat text.

In our model, we will focus on maximizing the F1 score to achieve a balance between Precision and Recall.

Post-Model Training
Post-model training, we need to evaluate the performance of the training bot. There are two methods:

- Human Evaluation: using crowdsourcing services to give feedback on the model's performance.
- Conversation Analytics: Once up and running, we can survey after a chat score to determine how well the chatbot performed.

In both cases, we can use the feedback to determine what corrective actions need to be taken for continuous model improvement.

Minimum Viable Product (MVP)

Design

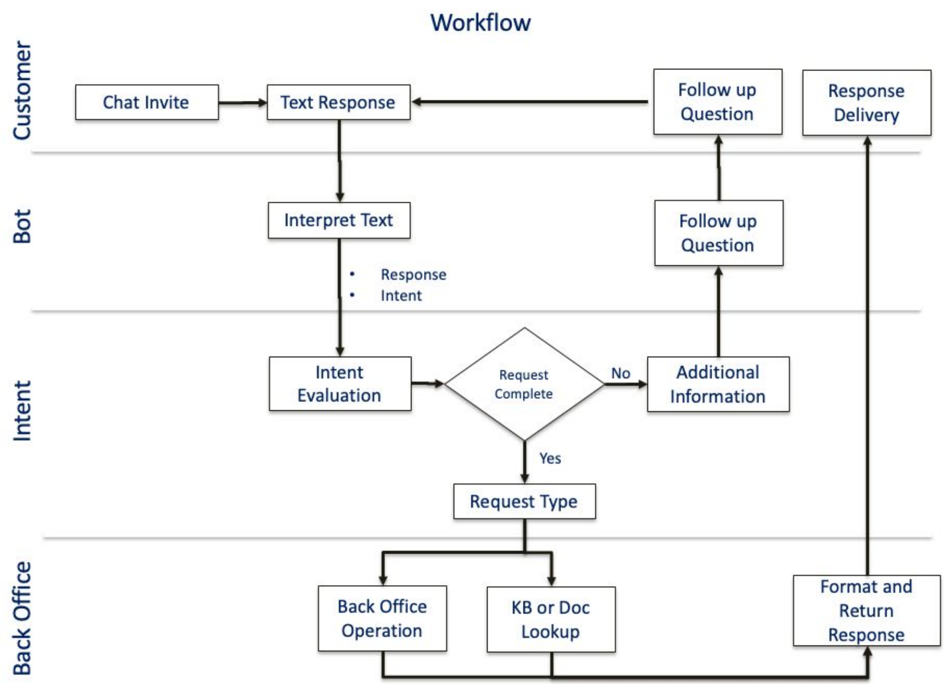
Chatbot MVP is defined by the chat conversations it can understand and the actions it can take. MVP would be based upon the following areas:

- Action-based:
    - Account Support. (password, order status, change of address, returns,

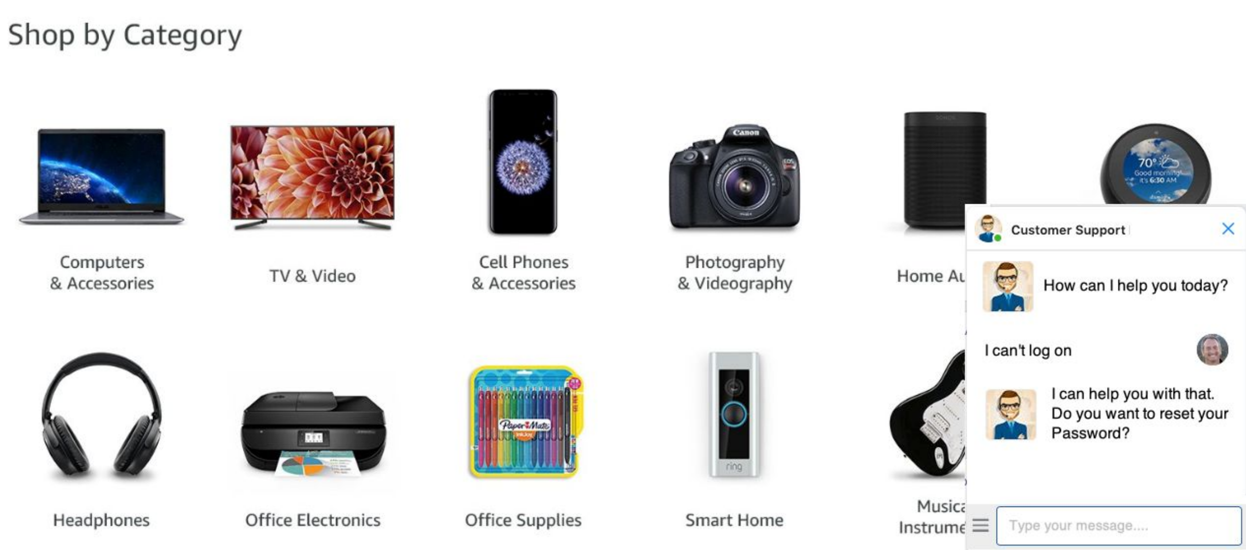
In the future we will add :

- Knowledge-based queries
    - Existing FAQs
    - User manuals and how-tos
- Sales agent functionality. This will be:
    - Context related, where on the web page
    - Perform up-sell / cross-sell
    - Product search and selection

The basic workflow of the product is as diagrammed below. Chat text would be interpreted by the bot and passed to the Intent Evaluation system. Here we will determine if we have enough information to process the request or do we need to add a follow-up question.



The Key UI will be the chatbot widget that will be embedded on each page of the eCommerce site. From a users point of view, there will be a popup that they can access from clicking on the chat prompt. Example:



Use Cases

The persona that we are designing for is an Adult Consumer. It is expected that they have experience using eCommerce sites like Amazon; thus they will have a high expectation for the user experience.

For account management, we can illustrate the functionality with three epic-level use cases for our Adult Consumer for our MVP launch.

Epic: MVP Launch of Account Management

- Story 1: Process an account password update
- Story 2: Get Order Status
- Story 3: Process Change of Address

Story 1: Process an account password update

1. User access the chatbot widget from the floating chatbot window
2. The chatbot will open with the question "I can help you with any account issues".
3. Consumer: "I need to reset my password".
4. The chatbot evaluates the request. Determines it is a password reset. The intent requires a username or password to process, so it needs to collect this from the consumer.
5. Chatbot responds: "OK, I can help with that. Can you give me your username or email address?
6. Consumer: "joeDoe@gmail.com".
7. The chatbot evaluates the response and determines if the email is valid.
8. The chatbot responds: "OK, I have sent the password reset to your email address. Can I help with anything else?
9. The consumer opens the email and uses the secure line to update their password.

Story 2: Get Order Status

1. User access the chatbot widget from the floating Chatbot window
2. The chatbot will open with the question "I can help you with any account issues".
3. Consumer: "I need to know when my order will arrive".
4. The chatbot evaluates the request. Determines it is an order status request. If the user is not logged on, the intent can be fulfilled by looking undelivered orders for the current user. If the intent requires more information, it needs to collect this from the consumer.
5. The chatbot responds: "OK Sue, I can help with that. I have two orders on file, one for electronics and one for clothes. which one are you looking for?"
6. Consumer: "The second-order".

7. The chatbot evaluates the response and lookup the information
8. The chatbot responds: “Your order for clothes will be delivered this Friday before 8 PM. Here is the tracking number ADED34345FB40234. Can I help with anything else?

Story 3: Process Change of Address
1. User access the chatbot widget from the floating Chatbot window.
2. The chatbot will open with the question “I can help you with any account issues”.
3. Consumer: “I need to record a change of address”.
4. The chatbot evaluates the request. Determines it is an account update request. Logged in users can be taken immediately to the appropriate page. If not, the chatbot needs to get the user logged in before the request can be processed.
5. The chatbot responds: “OK Bill, I can help with that. I need you to be logged in before I can do that. You can log in here. I can wait.”
6. Consumer: Goes to the Login link and logs on.
7. The chatbot: “Thank you for that. Here is a form you can use to update your address. Can I help with anything else?”
8. Consumer: Follows the link to update their address.

Roll-out

The rollout will happen in multiple stages. We are deploying on a live website with 10,000 users each day. The rollout process is to evaluate each rollout stage and correct any issues that arise.

Stage	Evaluation	Success Factors
Alpha	Crowdsource evaluation of the chatbot with testers giving a satisfaction and accuracy score.	Satisfaction rate above 80%
Select rollout / Beta	Rollout to select live customers. These are friends of the firm that will provide satisfaction and accuracy feedback	Satisfaction rate above 80%
A/B Testing	Evaluate the chatbot against a live agent. • Send 20% to chatbot (Challenger model)	Satisfaction rate above live chat satisfaction rates

	• Send 80% to live chat (Control model)	
Launch	The actual launch, to make available to all customers	Monitor and evaluate satisfaction scores

Post-Launch

The post-launch plan. The plan is to continue to evaluate the performance of the model and the customer adoption. Satisfaction scores of chatbot sessions will be used for the evaluation. For this we will use:

- Conversation Analytics: Once up and running, we can survey after a chat score to determine how well the chatbot performed.

If satisfaction rates are not close to live chat; we will need to pull the product and re-evaluate.

Post-MVP-Deployment

Designing for Longevity

To improve long term, Post-MVP, we will add features to support additional functionality:
- Documentation Lookup
- How-tos
- FAQ responses
- Sales Agent
- Regional / Language updates.

New features will require additional data sets and training of the model regularly. These additional datasets will train for new functionality.

Additionally, we want a continuous improvement process. We will continue to evaluate the performance of the chatbot and gauge satisfaction on post-chat surveys.

From the live chatbot we will continue to collect real-world data. We can measure the effectiveness of this data to create additional training specific to improving how the chatbot reacts to customer requests.

We will continue with a smart feedback loop to monitor the performance of our model. We can continue with A/B Testing, sending some users to live chat since this is the benchmark we want to evaluate against. We will evaluate the chatbot against a live agent as follows:

- Send 80% to chatbot (Control model)
- Send 20% to live chat (Challenger model)

With both evaluate chatbot dialogs and live chat transcripts, we can support the continuous improvement process by regularly updating the model with the new data to build a more accurate model.

Updates to the data will be rolled out as new versions on a regular basis. This is expected to be at least four (4) times a year. We will support industry-standard versioning processes and have plans in place to revert to old versions if needed.

Monitor Bias

The chatbot will be supporting the eCommerce experience, including account management and product research. Most of our training will be based upon real-world data. We need to be sure that as we clean and annotate data for training, and we have the annotators to flag cases of bias.

We also need to do sentiment training - that is, we need to train the system on how users are conversing in the chatbot. Are they happy or mad? We need to have different responses based on what attitude is detected.