Customer Service Chatbots and Natural Language

In customer service, everything essentially starts with a question:

- **"Where is my order?"**
- **"What is the status of my claim?"**
- **"What is your return policy?"**

A contact center agent has a lifetime of experience to leverage for efficient and accurate language interpretation. Agents are a limited and valuable resource, however, and this is why contact centers increasingly turn toward self-service as an option. It is also why software that can answer questions posed in natural human language – specifically, software in the form of a "chatbot" – has become such an attractive value proposition for contact center managers.

This white paper explores what is involved in teaching a computer how to deal with natural language. It introduces relevant concepts and discusses different technical approaches that are available.

Natural Language Processing (NLP), Understanding (NLU), and Generation (NLG)

The science of Natural Language Processing (NLP) is concerned with the task of performing computational tasks on natural, human language. This is not a simple task; human language is full of ambiguities, nuances, and complexities that are far from easy to model computationally. What does the sentence "We saw her duck" mean?

- That we have viewed a certain bird that belonged to a female person?
- That we observed a female person perform the action of ducking?

NLP also comprises many different subtasks; in the conversational space alone, we can divide it into Natural Language Understanding (NLU) – processing incoming text and deriving meaning from that text – and Natural Language Generation (NLG) – taking a meaning that is to be communicated and expressing that meaning in appropriate text. Besides these areas within the science, there are many others as well, from Sentiment Analysis – determining the polarity and topic of text that expresses feelings or opinions – to the complex task of taking text from one language to another in Machine Translation. Of these subfields, it is NLU that has recently been at the center of the work toward creating customer service chatbots.

Why do Bots Need Natural Language Understanding?

In the world of chatbots and Conversational User Interfaces (CUIs), one of the biggest challenges facing them is understanding the text produced by a human user. The reasons why this is difficult stretch across broad categories, but at the center of them is the infinite variability of natural language. There are uncountable different paths a user can take to express a single concept. In the face of this variability, a human's ability to extract meaning is something we may never completely duplicate, but science has worked toward this goal for decades.

The basic objective of NLU in a customer service chatbot is to determine the connection between incoming text from a user and the appropriate system response. This response can be to provide a simple answer to a question, or it could involve initiating an action that the user has requested, or it could be to store the information that the user has provided in response to a question prompt. Most approaches to NLU in chatbot toolkits therefore break down NLU into two subtasks: classifying **intent** and extracting **entities**.

Classifying the **intent** of a user's text is part of a user-initiated dialogue turn: the user has expressed a request, or a posed a question, and the chatbot needs either to classify this request or question as belonging to one of those it knows how to handle, or to reject it as a "no match" event. This will determine which of the actions or responses the chatbot will take.

- "I would like to book a flight from Boston to Phoenix" **>** FLIGHT BOOKING
- "Can I add a checked bag to my reservation?" **>** ADD CHECKED BAG
- "I would like to book a flight." **>** FLIGHT BOOKING
- "What flights are there from Orlando to San Francisco?" **>** FLIGHT BOOKING

Extracting **entities** from user text is typically part of a dialogue turn that has been (at least implicitly) system-initiated. The user is providing information in response to a question.

What is your departure city?

- "I would like to leave from **Boston**."

Or the user is anticipating a question and providing the information in advance:

- "I would like to book a flight from **Boston** to **Phoenix**"

In some cases this information could be a selection from a defined set:

Would you like to pay with MasterCard, VISA, American Express, or a different method?

- "**MasterCard** please."

In others, it takes a freer form:

What is your email address?

- "**Aimee123@myISP.net**"

The job of the extraction is to pick out the information from the text, and possibly normalize this information when (for example) there are multiple ways to express the same information:

What date would you like us to start holding your mail?

- Jan 3 2017
- 3 Jan 2017
- 3 Jan
- 1/3/2017
- 1/3/17
- 1-3-17
- Next Tuesday

Another term for this task of extracting entities from the user task to complete needed information for a given use case is **slot-filling**.

Machine Learning vs a Linguistically-Based Approach

A large number of solutions on the market perform intent classification using a model derived from a machine learning algorithm, and for very good reason. Approaches to conversational bots go back to the middle of the 1960s; most of these approaches were rule-based, where classification was based on lists of key words and explicit patterns of words that could be used to determine the user's intent. Capturing the diversity of possible ways to express a single intent required exhaustively listing the variations, which proved to be largely intractable task in a real-world environment.

With a machine learning approach, the rules for classifying user intent are not constructed explicitly by the developer. Instead, the classification engine is provided with examples of text belonging to each of the classifications. The engine then "trains" on these examples in order to determine the salient features to use when classifying a sentence as belonging to a particular intent. Once it has been trained, it can use that information to classify text in the future.

While this is a powerful approach that takes a significant burden off the developer, there are at least some potential difficulties:

- **You need data.** Machine learning algorithms are designed to be trained on hundreds of thousands of examples. While many implementations use a fully-trained general language model based on a language such as English which is then "bootstrapped" with the application-specific examples you provide, the fact remains that training on a small set of examples creates a less reliable model. However, in most cases an enterprise has either not collected data of this kind to work with, or has not cleaned and labeled the data with intents (a labor-intensive exercise). Generating "fake" data, unfortunately, can result in a model that does not match real users' patterns of expression. The result is a high rate of "I don't understand" responses, or the bot responding with the wrong answer.

- **The engine's ability to generalize from the data assumes that the data is sufficiently diverse.** This comes back to the fact that human language exhibits a lot of person-to-person variation; the model will only save a developer from having to exhaustively cover variations by hand if those variations have been adequately represented in the training data.
- **The result of training is usually not an inspectable model.** There is a lack of transparency that can be a concern in a customer service sphere; the customer care team usually has no way to inspect the model to verify that what it has learned from the data reflects the ways in which a human would classify the text. Likewise, modification of the model can be difficult. If text is misclassified, developers can add training examples to try and differentiate the intents, but they cannot directly modify the model.
- **The model cannot be generalized to other data.** This automatically-generated model reflects the data on which it was trained; it cannot then be used to classify the same intents expressed in a different language. It may not even work as well on text written by a significantly different user population, with different patterns of expression. In order to handle these other data, you will need to train new models for them. Starting from scratch to add a language can easily break the commercial viability of the chatbot concept as a whole.

An alternative to the data-driven, bottom-up approach is to revisit the top-down approach but with a more complex toolkit at your disposal. Aspect NLU is an example of this type of engine. It presents a generalized language engine that performs a complete linguistic and semantic analysis of a sentence. The developer writes classification rules that leverage this analysis, using linguistic abstractions rather than individual keywords. The variability of expression, and even differences between languages, can be represented in the language model rather than the rules themselves, making the task of coverage a more reasonable one. To specifically contrast it with the points above:

- You can start with less data. You do not need a training set, although obviously having example sentences to start with when building your rules will be invaluable. This allows you to get started more quickly even when your domain is large and distinctions between the different intents are complex:
  - "What alternative drives do you offer?"
  - "What engine types are there?"
    > same intent
  - "What hybrid drives do you offer?"
    > different intent
- **Diversity is built into the model.** A full linguistic ontology can empower you to work with abstractions rather than specific examples. These abstractions include:
  - The ability to capture synonyms automatically. A fully interlinked language model allows you to create rules in terms of the concept and not the specific word someone would use for it. This means that with a single reference, you can cover pop, soda, soda pop, and even tonic, without having to worry about whether your user is placing her food order from the Midwest or Boston.
  - The ability to use categories rather than specific words. For example, your rule might reference a vehicle and cover dozens of possibilities at once:
    - "I would like to insure my car." (truck, SUV, station wagon, BMW, …)
  - The ability to refer to multiple phrase structures at once. There are many different ways of structuring the same type of question – they should be easily referred to as a group, without having to provide examples of each one.
    - "How much does X cost?"
    - "What does X cost?"
    - "How expensive is X?"
    - "What is the price of X?"
    - "How much is X?"
  - **The result is an inspectable model.** When rules are built explicitly to classify intents based on this deep analysis of the sentence, then the team in the customer care department that is tasked with building the chatbot can inspect exactly what logic is being used to decide how to classify a sentence. If a sentence is misclassified, those rules can be directly modified. This is a capability that is not possible with the machine learning environments.
  - **The model is generalizable to other languages.** In an interlingual language engine, the rules are expressed in a form that is not specific to any language. Therefore, the rules apply equally to any language the customer uses, and the same initial effort can be

extended to globalize the chatbot with only small additional investment:

- "I would like two please."
- "Ich hätte gerne zwei."
- "J'en voudrais deux s'il vous plait."

Conclusion

Natural Language Understanding does not equal human language comprehension, with all of its subtleties and contextual dependencies; but in a well-designed chatbot, it can *mimic* comprehension fairly well. In constrained contexts, it can perform well at determining the user's objective and empower the bot to accomplish that task as a self-service interaction. Even a small rate of deflection can result in significant cost savings for the contact center, as well as providing instant responses with no wait or lag time from a *digital employee* who is available 24/7/365, never in a bad mood, never demands a pay raise, executes consistently, is only a "new employee" once, and is already fluent in multiple languages.

Corporate and Americas Headquarters
2325 E. Camelback Road,
Suite 700
Phoenix, AZ 85016
+(1) 602 282 1500 office
+(1) 602 956 2294 fax

Europe & Africa Headquarters
2 The Square, Stockley Park
Uxbridge
Middlesex UB11 1AD
+(44) 20 8589 1000 office
+(44) 20 8589 1001 fax

Asia Pacific & Middle East Headquarters
7 Temasek Boulevard
#08-02 Suntec Tower One
Singapore 038987
+(65) 6590 0388 office
+(65) 6324 1003 fax

About Aspect
Aspect helps enterprises break down the walls between people, processes, systems and data sources, empowering organizations to unite around the customer journey. Our customer engagement center offers native interaction management, workforce optimization and self-service capabilities that drive dynamic, conversational interactions and create a truly frictionless omni-channel customer experience. Leveraging all the benefits of the cloud and over 40 years of industry ingenuity, Aspect conveniently and easily connects questions to answers while helping enterprises keep service levels high and operational costs contained. For more information, visit **www.aspect.com**.