Making Chatbots Really Helpful and Smart with Ontologies, User Context, and Structured Content

By ALEX MASYCHEFF

RECENT SURVEYS and predictions from industry experts show that the vast majority of companies are planning to adopt chatbots before 2020. A big question is what these chatbots will be capable of doing and what additional value they will provide for the user. We've already seen chatbots that perform simple tasks, such as ordering pizza, booking an airplane flight, giving a weather forecast, and even providing basic customer support. But how well can chatbots understand users' needs and offer precise and useful advice tailored to a specific user's situation?

A Conversation with a Customer Support Chatbot

Imagine a company that makes printers. Let's assume the following.

- There are three models of the printer: EasyPrint Basic, EasyPrint Pro, and EasyPrint All-In-One.
- EasyPrint Basic and EasyPrint Pro are inkjet printers. EasyPrint All-In-One is a laser printer.
- EasyPrint Basic is connected to the computer via a cable. EasyPrint Pro and EasyPrint All-In-One are capable of both cable and Wi-Fi connectivity.
- For the sake of simplicity, let's say that only three issues may occur: the printer doesn't print, there's a paper jam, or the print is too light.
- If the printer doesn't print, it might be related to connectivity problems (for example, the cable is not plugged in or the Wi-Fi connection is lost).
- If the print is light, the cartridge or toner (depending on the printer model) may need to be replaced.
- If there's a paper jam, the troubleshooting procedure will be slightly different for each model; to retrieve the jammed paper from the printer, the panel you need to open will be different for each of the models.

I've built a simple troubleshooting chatbot for this scenario using a free chatbot platform and deployed it in Facebook Messenger. Here's how a dialog with the chatbot goes:

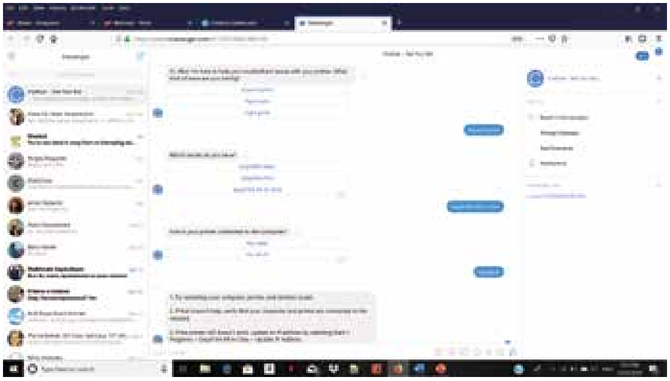

Figure 1. A printer chatbot conversation.

Is It Really Smart and Scalable?

This chatbot seems to be helpful. But how helpful will it be if the product and user's request become more complicated? Keep in mind that in this example, we are dealing with only three product models, and there are only a few variations in the troubleshooting procedure.

While the approach shown above is easy and relatively cheap to implement, it can only cover simple scenarios. In real life, the number of variations will be far greater, which means several things:

- Content varies depending on multiple parameters, such as product model, release, audience, or market, and these parameters can come in all kinds of variations. The complexity of button-driven navigation, like in the example above, will grow.
- With multiple parameters, button-driven navigation might be confusing, because users won't necessarily understand under which category the issue falls.
- Button-driven navigation represents a simple linear flow that shows only one perspective. As in the example above, it allows the user to determine what to do if EasyPrint All-In-One with Wi-Fi connectivity doesn't print, but it doesn't let you perform a reverse operation—to ask what kind of problems the Wi-Fi connectivity may cause. More precisely, the approach above would require designing a separate flow for each perspective.
- It's great when a chatbot answers the specific question the user asked. However, a challenge of the information age is that we don't know what we don't know. The user doesn't necessarily know which questions to ask to achieve a specific goal. Therefore, if the user didn't explicitly ask a question, the chatbot won't give an answer, while that information could be important and helpful for the user.

At some point, you'll need to allow the user to type free (and thus, unpredictable) text. The question is: how can a chatbot retrieve the user's goal, also known as the user's intent, from that question and map it to a piece of content that precisely addresses the issue?

There are a couple of major approaches:

- You can try to recognize the user's intent by analyzing keywords that appear in the user's request.
- You can pre-define utterances for each user intent, and then try to find a match between what the user typed and one of these utterances.

Analyzing Keywords

With keywords, you define a mapping between what the user types—specifically including a certain word or combination of words—and the answer that the chatbot should display as a result.

A problem with this approach is that one keyword might appear in different contexts. In other words, different answers should be given based on context. For example, if the user types, "I'm out of paper," the chatbot should probably provide instructions for how to order more paper. But if the user types, "My printer displays this message: 'Out of paper,'" the chatbot should probably provide instructions for how to load paper into the printer. Thus, keywords might work only for really simple scenarios.

Identifying Utterances

A more advanced approach for more complicated use cases is using utterances. An utterance is a way to express the user's intention. For example, if the user wants to replace a cartridge, there are many ways to ask about it. For example:

- How to replace a cartridge?
- How should I change an old cartridge to a new one?
- What's the procedure for replacing my cartridge?

A problem with utterances is that you have to define at least 15 utterances for every intent the user might have. Think about how many intents a user might have when working with a printer: print a document, set the print quality, connect the printer to a computer, align a cartridge, copy a document, print on the both sides of the paper, and so on.

Handling Incomplete Requests

To see how complicated the situation can get, add in multiple product models, and therefore increasing the number of content variations to be handled.

If this user asks, "How can I replace a cartridge," this request is incomplete. It doesn't provide sufficient information about the user's situation. In which model should the cartridge be replaced? Is it a black or a color cartridge?

The way this problem is traditionally solved is by using entities. You can think of an entity as a parameter of the intent. For example, in the question about replacing a cartridge, at least two parameters must be provided: the printer model and whether it's a black or color cartridge.

If any of these parameters is missing in the user's request, the chatbot needs to request this information from the user by asking a question. In the image below, one entity which represents the cartridge color is provided, but the second entity representing the model is empty.

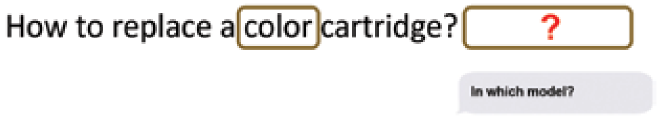

Figure 2. A missing entity in a user's question.

One problem with defining entities is the same as defining utterances: you have to specify intents, utterances for each intent, and then manually label entities in each utterance. It's a bit easier with machine learning, because with a sufficient amount of utterances, an entity-extraction engine can learn to automatically recognize entities. After manually labeling the entity in a number of utterances, this entity will be automatically recognized in other utterances. A lot of manual work has to be done, however, to train the entity extraction engine. To address these challenges, we can use a combination of three components: ontology, user context, and structured content.

Ontology

Ontology is a structural representation of knowledge about a domain. An ontology consists of concepts in and about the domain and relationships between those concepts. Ontologies are stored in a machine-readable format, so they can be processed by intelligent applications, such as chatbots.

This is how an ontology for our printer domain can be constructed:

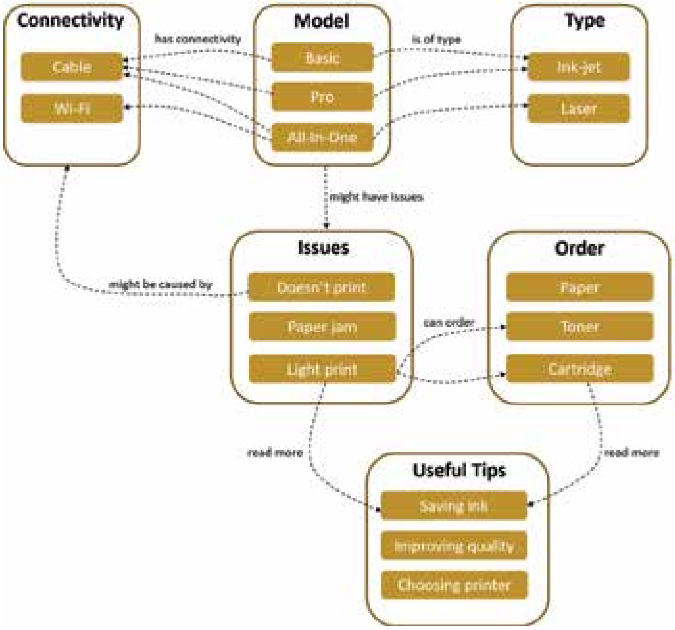

Figure 3. An ontology for the printer domain.

The ontology defines the types, model, and connectivity options, as well as possible issues that might occur. It also shows how each of these options relate to each other.

You can think of ontology as a *knowledge map.* In the same way that GPS navigation needs a geographical map to guide you from point A to point B, a chatbot requires a map to guide the user to the goal. GPS navigation needs information about where you want to go, whether you want to avoid toll roads, whether you prefer safety over speed, whether you want to avoid certain areas, and so on, and navigation through the knowledge map requires information about the user's needs, known as user context.

User Context

User context is the sum of all user circumstances at the current moment in time: goal, user's location, product, history of previous interactions, user's role, and language. The content that the user receives in response to a question is determined by the combination of all these and similar factors.

A user's context might come from many combinations of elements. For example, the user may have EasyPrint All-In-One connected to the computer via cable and experience problems with printing. Similarly, or the user may have EasyPrint Pro with jammed paper.

Structured Content

Because the user's context is granular, to match content to each specific user context, the content has to be granular as well. In addition, each content granule has to be labelled with information that corresponds to the situations to which it is applicable, such as product model, audience, and issues it covers. This is when structured content makes an entrance.

Structured content allows you to keep your content granular and to enrich it with semantic markup that contains information about the content. This semantic markup is readable by machines, and thus processes of retrieval and content assembly can be automated and become part of larger processes.

If we have an ontology *and* structured content, we can connect them with one another. This can be done in the ontology itself by providing ontology concepts with references to specific pieces of content. Alternatively, it can be done by adding to the content metadata that corresponds to the ontology concepts.

Navigation through Ontology

Let's say that the user asks, "How to get rid of a paper jam?" We still need natural language processing (NLP) to identify the user's intent. Once the user's intent is captured, a concept related to this intent will be identified in the ontology. In our example, it will be the concept that represents the paper jam issue.

By tracking the relationships for this issue, we see that it might occur in any of the three printer models. The user's request is incomplete, but we know what the chatbot has to ask based on the structure of the ontology.

Imagine the user responds to that question with "EasyPrint All-In-One." If there is no ambiguity or other relationships to be followed, we can identify a piece of content tagged with the corresponding metadata and display it to the user. Moreover, if the paper jam issue has a relationship with a concept representing useful advice, we can follow this relationship and display advice about how to avoid paper jams in the future. In this example, the chatbot generated questions automatically based on the structure of the ontology rather than on a manually built decision tree with multiple branches.

The ontology can also be used to provide the user with relevant information even if the user didn't explicitly request that information. Suppose that the print is too light. It might be an indication of low ink (if the user has Pro or Basic model) or low toner (if the user has an

All-In-One model), so it makes sense to offer the user information about ordering a new one. This can be done by defining in the ontology a relationship between the issue and the ordering information.

The ordering information, in turn, can be requested directly or in connection with other requests. Our ontology defines the relationship of the ordering information with other concepts in the domain. For example, if the user orders a new cartridge, some tips on saving ink can be provided.

Inferencing New Relationships

One more feature of ontologies is the ability to infer new information based on existing data.

Consider two statements:

- All humans are mortal.
- Socrates is a human.

What conclusion would you make out of these statements? "Socrates is mortal." This is new information, which wasn't defined explicitly. We discovered it by analyzing the existing data.

With ontologies, you can build rules on top of your ontology, and then let new information be automatically discovered.

Suppose we've defined the following rule: all laser printers in the EasyPrint family have Wi-Fi connectivity. Now you add a new concept to the ontology that defines a new model called All-In-One. You specify that the printer is a laser type and that it belongs to the EasyPrint family. Based on the rule we defined, it can be inferred that the All-In-One model has Wi-Fi connectivity.

There are many ways that inference can be used. It can help you reduce the amount of questions the chatbot asks the user. Some answers can be predicted based on the information already defined in the ontology or concluded from previous users' answers.

Inference also helps to provide the user with information that she didn't ask about explicitly but might be interested in.

The Value of Ontology

Now you might be thinking that defining intents, utterances, and entities requires a lot of manual work, but so do ontologies! Creating an ontology with semantic relationships and rules seems to require a lot of work, too. What's the difference?

Here are a few things to keep in mind.

Ontologies are not just for chatbots. Chatbot delivery is just one of many intelligent applications that can use an ontology. An ontology is a formal way to unify and standardize the knowledge that you have in an organization regardless of the format of the content.

Ontologies allow you to associate concepts defined in the ontology with relevant information resources, whether it's an article in an online knowledge base, a DITA topic, a Microsoft Word document, or a video. In other words, using an ontology, you can link a semantic model of the domain with the knowledge about the domain, regardless of the specific format in which the knowledge is stored, because when the ontology relationships between concepts are defined, the content associated with these concepts also becomes interconnected.

Therefore, an ontology can become a platform- and repository-independent knowledge model that connects department-level systems and repositories and provides navigation through the knowledge across the enterprise—*beyond* the boundaries of a specific department. The availability of machine-readable formats for describing ontologies (e.g, RDF XML, OWL, Semantic Web Rule Language) make them very suitable for this role, because various interfaces can be built for navigating through the ontology, including chatbots and customer portals. They can allow a content consumer to start from an instructional video created by the training department and get to best-practice advice through the web of semantic relationships defined in the ontology.

Because ontologies allow for machine processing, each team can use their own tools and business logic to process it.

In addition, the ability to automatically discover new relationships based on the existing information that is provided by ontology-related technologies allows you to find connections that might not, at first, be visible. With the constantly growing amount and complexity of information that teams and companies and users must deal with, this capability becomes critical.

Conclusion

If you want to provide your customers with accurate and precise information tailored to their specific needs, structured content should be an essential component of your solution. However, *structured content alone is not sufficient.*

That structured content needs to be powered by an ontology that represents a knowledge map. This map will allow you to dynamically construct individual pieces of structured content into an interconnected web of knowledge. And to navigate through the knowledge map, you must capture the user's context.

A combination of structured content, ontology, and user context will provide an information infrastructure that can help you guide the customer through the knowledge relevant to her needs. i

ALEX MASYCHEFF (alex@intuillion.com*) is the CEO of Intuillion Ltd. which develops solutions helping companies create and deliver personalized product information to existing and prospective customers. Alex has been in the content industry for 20+ years. He lead implementation of XML-based solutions in many companies, including Kodak, Siemens, Netgear, and EMC. Alex believes that using a combination of structured content, ontologies, and structured content, companies can provide users not only with precise answers on direct questions, but navigate them through the opportunities that the users don't even realize that they exist.*