

# CHATBOTS GONE WILD

CHAT BOT

The data science behind  
making sure your chatbot  
is worth talking to

CHAT BOT

 **CrowdFlower**

# WHAT IS A CONVERSATION ANYWAY?



Our lives are full of conversations. From a text chain with your boyfriend planning dinner that evening or ordering an egg sandwich to the bodega to a quarterly business review or a heated political conversation over cranberries at Thanksgiving, they come in every shape and size, varying in topic, importance, and convention. But how often do we actually think to ourselves: "Why am I having this conversation? What's the point?"

In this eBook, we take the perspective that conversations are about coordinating joint action. This squares pretty closely with the work of Stanford psycholinguist Herb Clark. Conversations are dynamic, full of signals and interplay. They can go exactly as we expect or end up places we never thought they would. "Chatbots" and "conversation" are nouns, but to get things right, we've got to think more in terms of verbs. How do we coordinate with each other? How do we establish the common ground of what we're talking about it?

# WHAT IS A CONVERSATION ANYWAY?

A huge part of this is convention. That is, the next time you're walking down the street, imagine trying to ask someone else to swing dance. A random pedestrian understands the conventions of "excuse me do you have the time" or a simple head-nod-of-hello from a stranger, but a music-less dance on the sidewalk? That's unconventional. You're going to get an unconventional response. And it's probably going to involve a lot of furtive looking away.

The same should apply for a chatbot, of course. Whether you're talking to a travel agent or an airline chatbot, the conventions mean they should be able to book you a hotel or change your flight. It's going to be a lot harder to get them to talk to you substantively about politics or Euclidean geometry.

Now, convention is a big part of having a successful conversation, but what about the substance of that conversation? Generally, we can break these up into four main components:

## 1. Reciprocal greeting:

This one should be pretty self-explanatory, but when I say "hi," you say "howdy." There are of course myriad flavors of greeting, but the goal is the same: to establish rapport.

## 2. Transfer of information:

Think "what are you doing this weekend?" and its answer. Conversations are about giving and receiving information.

## 3. Instigation of behavior:

Think "let's get a cup of coffee tomorrow" or "can you help me with this?" Part of talking to others is making plans and requests.

## 4. Settle on a viewpoint:

Think "I agree, artificial grape flavoring is the better than real grapes" or, really, any debate (as well as plenty of less charged conversations).

Admittedly, this is a bit of a clinical view, but it's important to understand this if you're going to create a chatbot. You need an AI that understands those discrete components and the interplay between them. But above it all, keep in mind the point of conversations: coordinating joint action. A chatbot needs to take input from

a user and coordinate what's happening next. Is your company issuing a refund? Do you need more information from a user before you can continue? Does a real person have to step in and take over? A chatbot that can't answer these sorts of questions isn't going to be much good for you or your customers.

## WHY BUILD A CHATBOT ANYWAY?

If you think chatbots are a bit niche, that's understandable. There's been a lot of breathless hype around chatbots and other types of digital assistants and you're right to wonder "what's the big deal anyway?" After all, it's going to take some work and investment on your end before you see any value. And it's smart to make sure the value you'll see is worth the effort you're putting in.

So why do it? There are a few salient reasons.

For starters, most businesses build chatbots to help with customer service. That can be for anything from helping a prospective shopper find out if her size will be available next week to rescheduling a hotel booking to dealing with more complicated, sensitive financial issues.

Here, it's important to remember that when chat became a typical part of customer service operations, businesses were able to save a lot of cost on a per customer basis. After all,

customer service reps on chat can help more than twice the amount of people per hour and most estimates put the cost per interaction at about 30% of phone interactions. With chatbots, those numbers can get even better. Because most issues are fairly commonplace and predictable (a hotel booking site knows



that conversations around cancellations, room upgrades, and check-in times will be a numerous), in many cases, chatbots can handle a majority of the issues your customers have. This frees up your professional CS reps to do more. Instead of dealing with cookie-cutter issues, they get the complicated stuff bots can't handle or ones that specifically need a human touch.

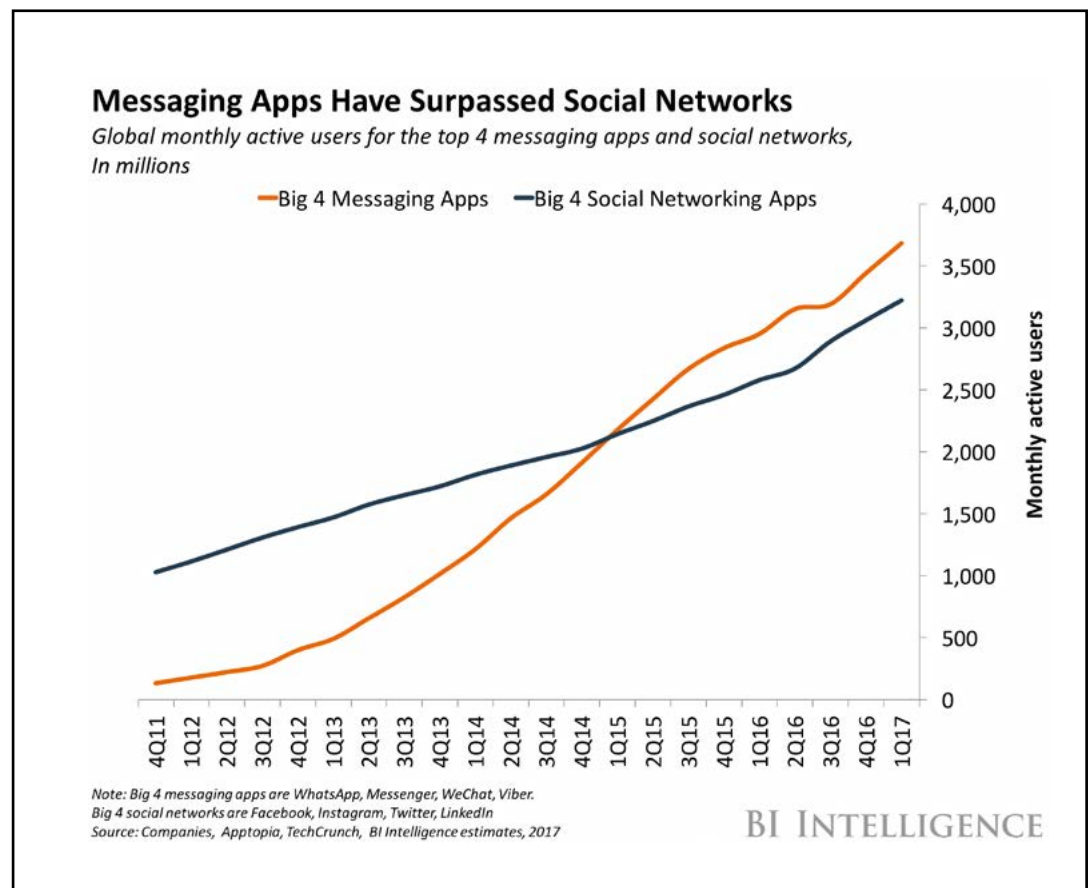
Stanford linguist Chris Pott's has a rule of thumb for exactly this: "Normal events are reported with normal language. Unusual events are reported with unusual language." Chatbots deal well with normalcy. It's the unusual they struggle with.

That's precisely what your human representatives can—and should—handle. By funneling each conversation to the optimal agent (phone, chat window, or chatbot), you'll save your customer reps from monotony and you save some money. Moreover, chatbots are available 24 hours a day. They work on Christmas. They don't call in sick. They're

reliable. Sounds like a good deal.

Now, here's a bit you might not know: messaging apps have skyrocketed in popularity. In fact, according to a Business Insider, they're now more popular than social networks. And that popularity gap is actually widening.

In other words, messaging apps are where your customers are. And chatbots live naturally in these apps. If your customers are spending more time in WhatsApp and Messenger than they are on Twitter and LinkedIn, why not take advantage of that? Chatbots are a natural fit here. They give your customers a place to talk to your business when



it's convenient for them without the (admittedly minimal) hassle of downloading a special app or browsing your mobile store or calling in and navigating a phone tree.

So why chatbots? They save you overhead, they rescue your employees from monotony, and they interact with your customers on the apps they're using already.

## OKAY, I'M CONVINCED. SO NOW WHAT?

Alright. Let's build a chatbot. The first question we want to answer is why we're building one. The most common reason is the one we used above, namely, building a bot that interacts with customers. It can help with buying decisions in retail, hotel bookings in travel, trouble shooting for large SaaS providers, or really anything that requires a lot of employees to coordinate joint action with your customers.

Remember, this is about goals and it's about context. You don't need to build Siri here and, well, if you've tried to use Siri for nuanced tasks, you know that it's not altogether that successful. Find metrics you're looking to shift, whether it's customers served per hour, NPS score, you name it. You'll want to track those and smart bots should give you substantive gains across a wide variety of important measures.

For this eBook, we're going to use the example of an airline chatbot to show you a few of the things you can do to make your bot smarter,



more agile and robust, and, fundamentally, successful. Of course, these aren't the only sort of bots there are. Chatbots are on the market now that do everything from forecasting the weather, keeping you up to date on news, scheduling your meetings, helping you manage your money, or even being your friend if, you know, you're into that sort of thing. But the techniques we'll describe here are applicable to all these bots. They all aid in conversation and, after all, that's the purpose of a bot in the first place.

Once you've got your goal, the next thing is working out what normal interactions you can count on. One of the dirty little secrets around chatbots is that a lot of the answers are preprogrammed by engineering teams. For example, take something like what happens when you ask Siri to tell you a joke. "She" isn't exactly going to spit out a joke she heard at a party last week. Rather, Siri is consulting a lookup table. The Apple engineers expected we'd ask so they coded it in. And for a lot of companies, this really works! Remember how we mentioned the notion that "normal events require normal words" above? Well, you'll likely be hard-coding some normal responses to normal events. The trick is figuring out what's normal interaction for your customers. Odds are, your reps will know things or your ticket tagging system, search logs, or a whole host of other big data sources will give you the information you need to get going.

So if we're just hard-coding responses, isn't that easy? Shouldn't there be chatbots working everywhere? Well, it's not exactly easy. What you're talking about there is simply an information retrieval system. In other words, search. But successful chatbots aren't search bars. They need to have interactions. They need to have conversations. They need to coordinate joint actions.

**Successful chatbots aren't search bars. They need to have interactions. They need to have conversations. They need to coordinate joint actions.**

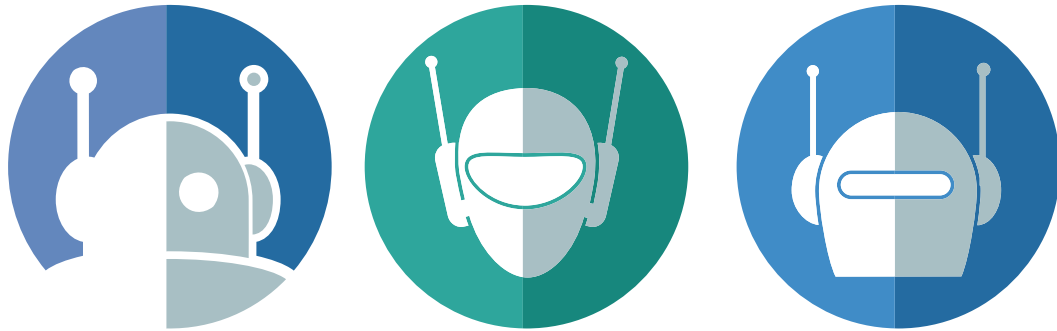
There are four types of algorithm training we want to look at here. Odds are, you'll need to get training data from your database or from a platform provider that can supply you with the sort of robust data you need for your chatbot to understand and converse with customers. In other words, for it to be worth your time.

We'll explain each on a high-level, then examine each in detail over the remainder of this guide:

- **Utterance:** How many ways are there to say the same thing? Your chatbot needs to understand as many as possible or it's going to be perpetually confused.
- **Relevance:** Is a particular response relevant to a particular question?
- **Intent detection:** Does your chatbot understand what your customer is asking? It can't coordinate joint action if it's unsure about intent.
- **Entity extraction:** There's a difference between "this burrito is so bad" and "I want a burrito so bad." Entity extraction is a good way to teach an algorithm the subtleties of language.

There's one last question before we get to creating a corpus of questions, tuning your chatbot, and the more nuanced parts of building a conversational AI: what's your bot's personality?

## IS YOUR CHATBOT NEO OR OBI-WAN?



So you're creating a chatbot. You know who else creates non-existent people who don't talk? Screenwriters. Moreover, they create conversations. Characters talk so they can coordinate joint actions.

Interestingly, in movies, a lot of conversations serve another purpose: letting the audience in on what's happening. Dialog needs to help us understand what's going on in a film. It's a bit different than everyday speech. But when we actually dig into the data around characters (here, we'll look at a collection of 9,035 different characters in 617 movies), we start to see a

pattern that's worth bringing up here.

Take someone like Neo in *The Matrix* or Will Smith's character in *Men in Black*. They are thrown into really different realities and have to try to get a hold of what the heck is going on, so they ask a lot of questions. In fact, they ask more questions than making simple statements. On the other end of the spectrum? Obi-Wan Kenobi, Terminator, and Spock.

So what's that pattern? Characters who want to figure out what's going on ask questions. Aliens and space wizards don't. Which brings us to chatbots. Chatbots aren't sagacious Jedis. They need to

How long can you talk to a character before they ask you a question?

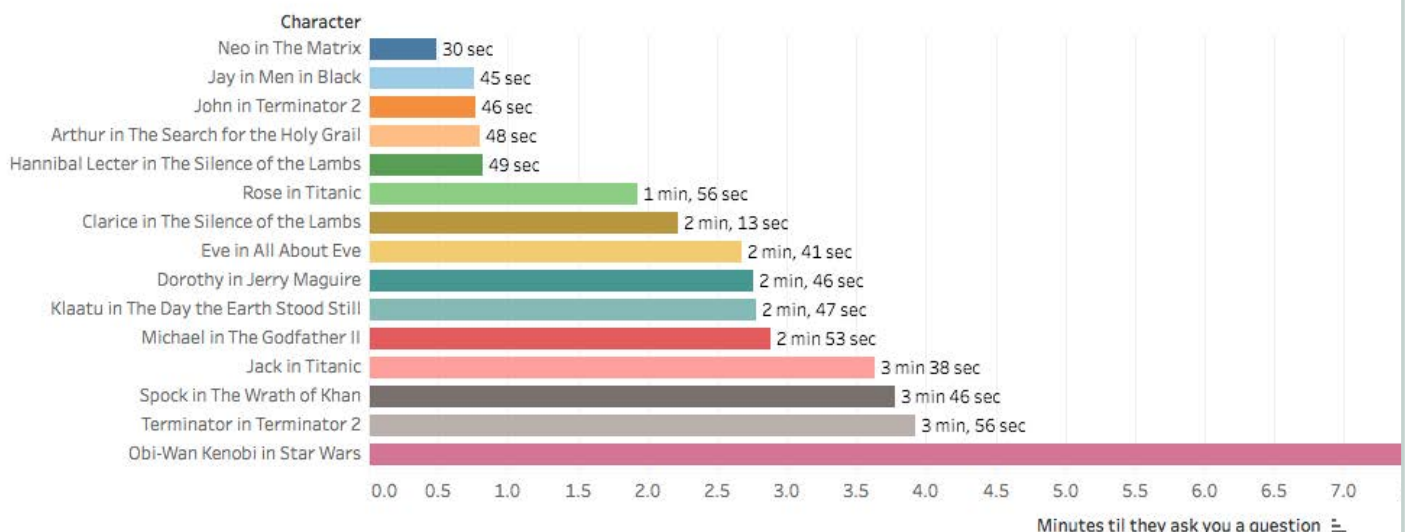




figure stuff out, work through problems, orient themselves in a conversation, and coordinate joint actions. In fact, chatbots that ask a lot of questions tend to perform better.

Take a simple phrase like "Book me a flight to New York next Wednesday." Are you willing to let a chatbot just go off on its own here? Absolutely not! There are tons of questions that you need answered to optimize this conversation. Do you want to fly in the morning or afternoon? LaGuardia or JFK? How about layovers? Do you have frequent flyer miles you want to use? An extra bag?

You get the idea. A chatbot that asks those questions acquits itself like someone who wants to understand. A considerate chatbot or human is also attuned to what sort of follow-up questions are important to get early versus nice-to-haves-

that-just-annoy-people. A customer's goal may be to book a flight, but they may also be in a rush. We often pursue multiple goals.

Past that, remember that you'll be hardcoding a good amount of responses. How do you want your chatbot to behave? Most companies have a personality. They're fun or playful or serious or sarcastic or, really, anything a person can be. Make sure your chatbot mirrors the personality of your brand or, at the very least, is polite and helpful to the people interacting with it. If you're building a customer service bot, build it like your ideal rep, who may be feisty but probably isn't mouthy.

Now? Let's get building

## THE FOUR TYPES OF LANGUAGE TASKS FOR TRAINING A CHATBOT

### 1: Utterance, or, How Many Ways Can You Order a Pizza?

To work at all, your chatbot needs to understand what users are asking it to do. And while you can likely easily identify the most frequent, most normal requests from a user, it's tough to come up with every permutation of those core questions on your own.

That's what utterance data collection is all about. The task is simple: set up a task where a bunch of people come up with different ways to ask the

same question. What's the question? That's up to you and your team. But you'd be surprised just how many ways there are to ask for the simplest things.

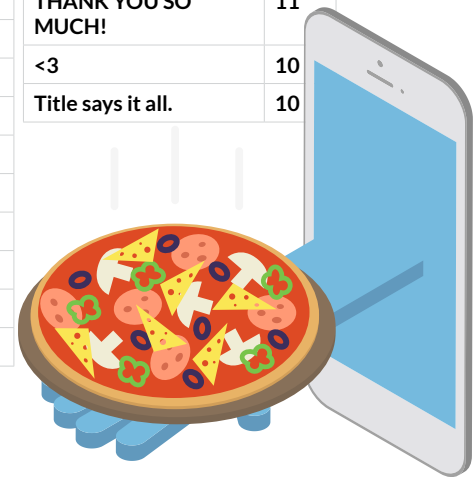
### KEY USE CASES

- Transforming FAQ content into a chatbot (you've already written answers, but want to match them to lots of different questions)
- Building up voice/text activation for a new feature (how many ways are there to ask for a song to play?)



An example? Reddit's Random Acts of Pizza, where people ask for pizzas and potentially the community responds. If we look at 5,671 requests for pizza, we'll find that 99.4% of all of them have unique titles. In fact, there were only four repeats at all! Inside the body of the posts themselves, the only repetition that exists over 27,000 sentences are basically just greetings and assorted gratitude:

Thanks!	122	Thanks guys!	15
Thanks.	69	Thanks :)	13
Thank you.	55	Thanks so much!	12
Thank You!	41	Hey guys.	11
Thanks for reading.	37	THANK YOU SO MUCH!	11
Thanks	34	<3	10
Thanks for reading!	28	Title says it all.	10
Thanks in advance.	22		
Thanks in advance!	21		
**	19		
P.S.	19		
Thank you	19		
Hello!	16		
Hi!	16		



This a good example of the breadth of just simple requests. "Please pass the salt" and "Salt!" are both ways to make a request, after all, but they feel rather different. And while people will interact with chatbots differently than people (think about how you search for shoes or use Google; it's not exactly how you talk to your friends), accruing a database of the ways people ask for things gives your chatbot fuel to answer those requests in kind.

Now, a section or two ago, we mentioned that we're going to use this eBook to demonstrate

how to create the data you need to train a chatbot. We chose to create data around an airline customer service chatbot, but of course, you can do utterance tasks for whatever utterances you want to capture.

For our example job, we chose to ask for ways to ask for "can I change my flight?" Again, there are no specifics here (like "I need to change flight 563" or "I have to fly to Vegas instead") so the pool of utterance data is artificially limited a bit, but here's how you do it:

This is your scenario:

### You want to cancel your flight

1) What might you say to or ask a customer service agent in this scenario? (required)

ⓘ Responses must be in English and relevant, or will be rejected.

2) What is another thing you might say or ask? (required)

ⓘ Responses must be in English and relevant, or will be rejected. Responses must be different then the first.

3) What is one other thing you might say or ask? (required)

ⓘ Responses must be in English and relevant, or will be rejected. Responses must be different then the first or second.

Pretty easy right? Now, one of the things that CrowdFlower prides itself on is quality control. But with utterance tasks, that can be tough. You can't come up with the "correct" ways to ask this question (in fact, you're trying to accrue just that data) so you can't use the typical test question format most of jobs take. We get around this in a pretty simple way: two different, intertwined jobs.

In the first, ask the crowd for ways to say a thing (in our case, how to change a flight). In the second, you present data labelers with those responses and ask "is this a legitimate way to ask that question?" It's that easy.

Next, let's look at relevance:

## 2: Relevance, or, Are We Making Sense or Not?

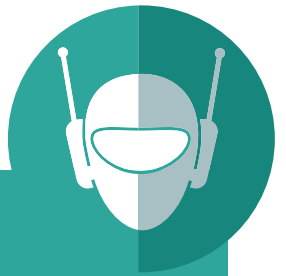
Once you have a set of utterances, you want to be able to match them with answers and actions. Relevance tasks do this by giving you training data about you can use to map utterances that users might say to the help pages and action-triggers in your database. They are usually of the form, "here's a question, here's an answer, how relevant is it?"

In doing this mapping, you are likely to find that certain flavors of questions need longer or shorter responses. The more a response just looks like "the best matching paragraphs" or "an adjacent answer from our FAQ section," the less direct help it offers, the less human it feels, and

the less satisfied your user is.

To get a sense of how people know what to say, let's look at the four maxims Paul Grice developed that people follow when talking. If you flout these maxims, things get weird.

1. **Quantity:** be as informative as you possibly can and give as much information as is needed, and no more
2. **Quality:** try to be truthful and don't give information that is false or that is not supported by evidence
3. **Relation:** try to be relevant and say things that are pertinent to the discussion
4. **Manner:** try to be as clear, as brief, and as orderly as you can in what you say and avoid obscurity and ambiguity



**A quick note:**  
the data from our example job is all available on our **Data for Everyone** page.

You can find that at [www.crowdflower.com/data-for-everyone](http://www.crowdflower.com/data-for-everyone)

We can reduce these even more. For Dan Sperber and Deirdre Wilson, the central thing is “Be relevant”. Or more formally:

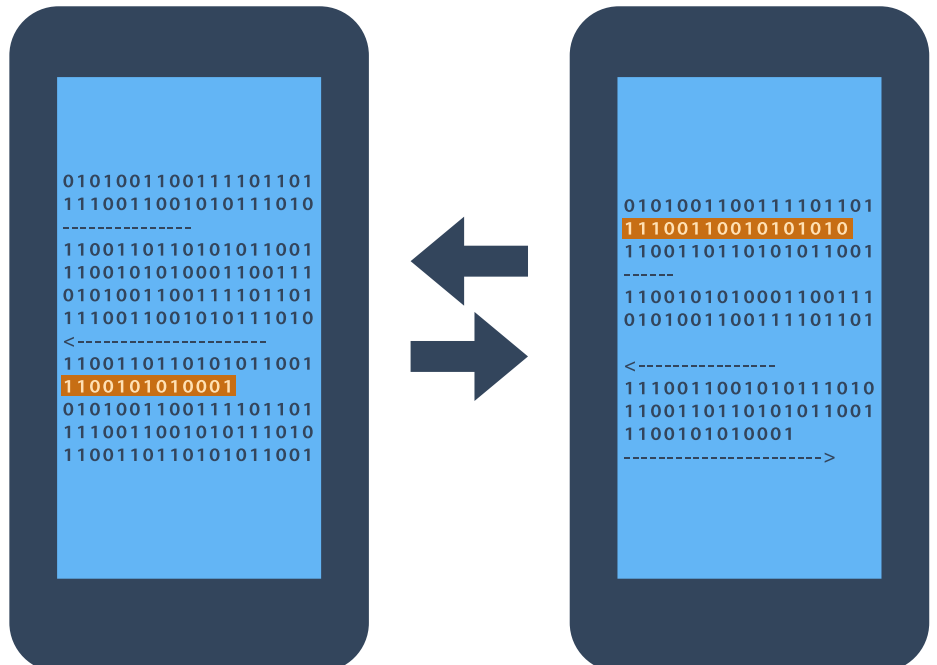
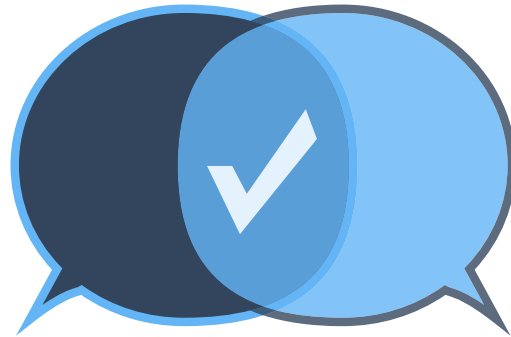
The issue for chatbots is they can have trouble understanding context. They're certainly worse at it than we are. And because of that, some of their responses are, well, irrelevant. And irrelevant responses make for bad conversations. They don't coordinate joint action.

This is one of the reasons it's much simpler to create a chatbot to handle discrete issues (like rescheduling a flight) than one that just wants to talk about any old thing.

You see similar tasks in search relevance projects: given a query, does this result match? Is it relevant? Doing that with chatbot question/answer pairings gives you the tools you need to tweak your models and make them more accurate. It also will show you where your model is falling down and where it's succeeding.

Every piece of communication conveys a presumption of its own optimal relevance--that is, it assumes it's worth your time to decode and it's as relevant as it can be given the communicator's abilities and preferences.

DAN SPERDER & DIEDRE WILSON



### 3: Intent, or, What Were You Trying to Do Anyway?

When we're engaging with people in joint activities like conversation, we are (or become) attuned to their intentions. That's what's behind the comedy of something like Lucy and Charlie Brown's "I know you know I know you know" chains of reasoning. Other minds aren't entirely opaque to us, even if we tend to fill them in with our own projections.

Much like the last example, you see intent work in informational retrieval projects like internal search relevance tasks. Basically: does this output match the intent of what someone wanted? When someone searches for an iPhone and they're presented with an iPhone case, does that match their intent? The same is true for chatbot replies. Given a question from your utterance corpus, how relevant is the answer your model or hardcoded bot returns?

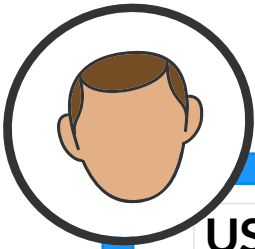
The reality is that relevance isn't quite enough for chatbots. Conversation is simply too complicated for simple relevance to make chatbot responses good enough.

Take the airline customer chatbot we're building. Imagine a customer typing "baggage fees?" What do they actually

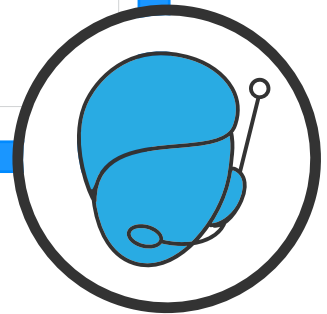
mean? Are they asking what the baggage fees for a particular flight are? Are they demanding a refund for baggage fees they were recently charged? A chatbot who doesn't understand context and intent might just send the customer to an FAQ about baggage fees. And that customer isn't going to be particularly enthused about that interaction.

Intent and relevance are intrinsically linked. You want to start the process by identifying which flags your chatbot will be able to support. Do you want to handle your top ten issues? Top five? You want to tackle as many permutations of those conversations as possible in your relevance and intent tasks. And keep in mind, these tasks are sometimes even more valuable for tuning your bot after it's been released or with test conversations you conduct with it. You'll be able to analyze whole conversations, find out where they fall down, and give annotators fuller conversations to understand customer intent.

Because, really, that's an important point here: intent shows itself most clearly in the context of a full conversation. That "baggage fees?" comment means a much different thing based on particular, individual conversations.



<b>USER:</b> What's this additional charge on my card?	<b>BOT:</b> You checked an additional bag.
<b>USER:</b> baggage fees?	<b>BOT:</b> Yes, that is the reason for the charge
<b>USER:</b> Do I get any amenities on this flight?	<b>BOT:</b> Complimentary wifi and drink service. You can pay for a hot meal as well.
<b>USER:</b> baggage fees?	<b>BOT:</b> Only for your second checked bag.



Intent tasks often present annotators with conversations (or snippets thereof) and ask users if the chatbot is understanding the intent of the customer. In the places it did not, it's important to understand where and why your bot hit a snag. Once that's understood, you can hone your models or hard-code answers to deal with precisely those issues.

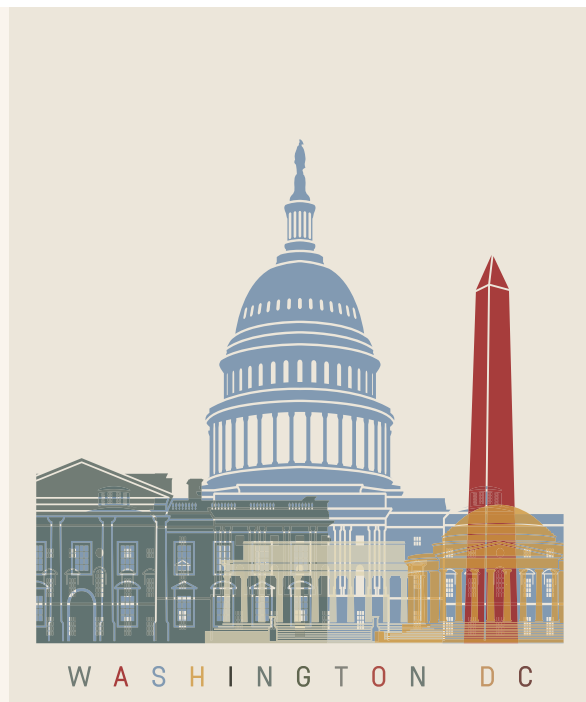
Last thing: remember that point we made about your chatbot's personality? That plays here. If your chatbot isn't sure it's going to

be relevant (essentially, it's unconfident about output) or is at sea over intent, just ask! Chatbots that deal with requests by asking a series of probing questions to find the exact thing that user is looking to do are far, far more successful than those that make pseudo-guesses where they're not fully confident. When in doubt, your chatbot should aim towards further clarity, not action.

#### 4: Entity Recognition, or, Which Washington is this Washington?

Entity recognition is the last major training job for your algorithm. Essentially, it involves looking at passages of texts and identifying "entities" within. Those might be places, people, product names, you name it, but generally work best looking for specific entities that are valid for your particular use case.

Take our example use case of an airline service chatbot. If you tell it that you're looking to go to Washington, what does that mean? Because it could mean any of the following:



- Ronald Reagan Washington National Airport, in Arlington Virginia
- George Washington International Airport, in Houston Texas
- Seattle-Tacoma Airport, in the state of Washington
- Any other airport in the state of Washington

You get the idea. Now, if you're building a chatbot that's looking to engage over American history, Washington has a totally different meaning. Ditto to a bot looking to give out college sports scores. The list goes on.

For starters, this is why more generic, multi-purpose bots are so difficult and why context is so important for any chatbot. But it's also why you need to work on entity extraction for your chatbot project. In fact, named entity recognition is one of the basic building blocks of natural language processing and it allows your bot to function properly.

Entity extraction projects require a little more attention to detail than simple intent, utterance, or relevance tasks, but they really power chatbots to be as valuable as they can be.

We've created an entity extraction tool that's very similar to a popular one you may have heard of called BRAT. Essentially, on our platform, you provide

users with text blocks and they highlight the entities you care about. You can see an example below:

U: my flight is late  
 B: Thank you for writing in.  
 B: Can you tell me more about your flight?  
 U: sure  
 U: hold on  
 B: Thank you.  
 U: just shows "delayed" but no reason  
 B: What is the flight number please?  
 U: goes to london at 725

**Step 2: Extract the entities from this excerpt from the text**

goes to london at 725

Flight number Location Airport Flight time Frequent flyer number Date Landmark Brand name Proper name

**Locations (required)**  
 725:18:Flight time

Check there is not an entity in this text

In that screenshot, we're interested in a few salient things to build to our airline chatbot. Note especially that numbers are important here. Is it a flight number? An arrival time? An amount of ounces for carry-on sunscreen? The more examples of named entities your model sees, the more it learns to understand that some time people typing

won't write "7:25" and instead just write "725" but your bot will actually understand. That increases your bot's accuracy, its ability to actually converse, and, yes, makes it function in the way it's supposed to: coordinating joint action.





Nice as it would be, you can't just buy chatbot software out of a box and simply deploy it. You need to test, tune, and train your chatbot. Hopefully, this eBook gave you the understanding of how that's actually done. But we do want to highlight a few of the key takeaways we'd love to leave you with now that we're finished:

- Conversations are about coordinating joint action. The best chatbots have real conversations and, thus, coordinate real joint actions.
- When in doubt, make sure your chatbot is curious. A curious chatbot understands what a user really wants before acting. And people are much more willing to answer a few extra questions than deal with bad outcomes.
- There are four major chatbot data projects. Each are important. They are:
  - Utterance tasks: How many ways are there to say a thing?
  - Relevance tasks: Does this response even make sense?
  - Intent tasks: What did the user want to happen here?
  - Entity extraction: What are these particular words exactly?

**If you'd like to talk to CrowdFlower about their toolset for chatbot training, please do reach out. We'd love to help.**





[www.crowdflower.com](http://www.crowdflower.com)

#### **About CrowdFlower**

CrowdFlower is the essential human-in-the-loop AI platform for data science teams. CrowdFlower helps customers generate high quality customized training data for their machine learning initiatives, or automate a business process with easy-to-deploy models and integrated human-in-the-loop workflows. The CrowdFlower software platform supports a wide range of use cases including self-driving cars, intelligent personal assistants, medical image labeling, content categorization, customer support ticket classification, social data insight, CRM data enrichment, product categorization, and search relevance.

Headquartered in San Francisco and backed by Canvas Venture Fund, Trinity Ventures, and Microsoft Ventures, CrowdFlower serves data science teams at Fortune 500 and fast-growing data-driven organizations across a wide variety of industries.

For more information, visit [www.crowdflower.com](http://www.crowdflower.com).