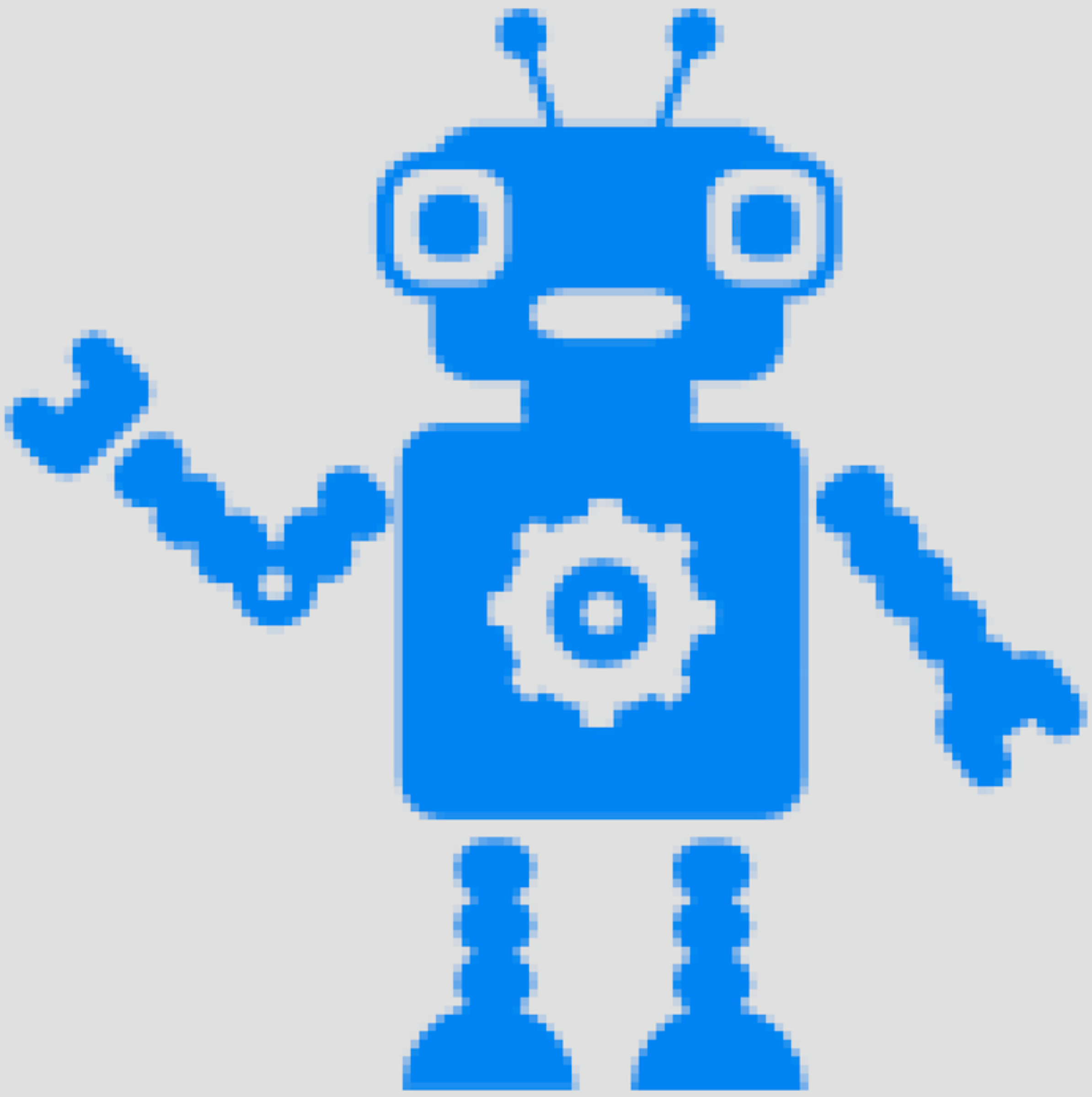
10 features all API.AI developers MUST know

Aravind Mohanoor



PREFACE

As someone who regularly answers questions on the API.AI forum, one of the things I repeatedly notice is how people seem to trip up over the same few basic features many times. This sometimes means they wait for hours and even days to get the answer, only for the answer to be a simple “Oh, you can use events for that”.

I have written this guide to help you avoid wasting your time over trivial features. This is not meant to be a reference volume which has answers to every question. In fact, it is the exact opposite. By learning about these features in API.AI, you will be following the Pareto’s principle of learning about the 20% or so concepts which will give you the most bang for your learning buck.

I also hope that reading this guide end to end should not take you more than a couple of hours, and I certainly encourage you to skim it and get to the end as quickly as possible, just to see what kind of features you have in API.AI, and then come back and re-read as necessary.

Aravind Mohanoor
June 2017

CHAPTER 1

IMPROVE PATTERN MATCHING

Based on what I have seen on the discussion forums, the template mode is one of the most powerful but obscure feature in API.AI. When people "discover" the template mode, they are frequently surprised. And at the same time, a light bulb seems to go on when they realize how many things the template mode is perfectly suited for. I went through a similar process too when I first learnt about it.

In API.AI, you can have two kinds of user says input. As you input some possible phrases, by default you will be using the "Example" mode, where API.AI will receive a few sample phrases and will be able to infer similar phrases by itself. This is great, until you require a little more fine grained control. Or sometimes the example mode isn't quite up to the task for a certain kind of phrase you wish to match. You will see a good example of this in chapter 4 which discusses handling complex user input.

In those cases you can use the Template mode. Enabling the template mode is straightforward. You just need to click on the @ sign in front of the user says.

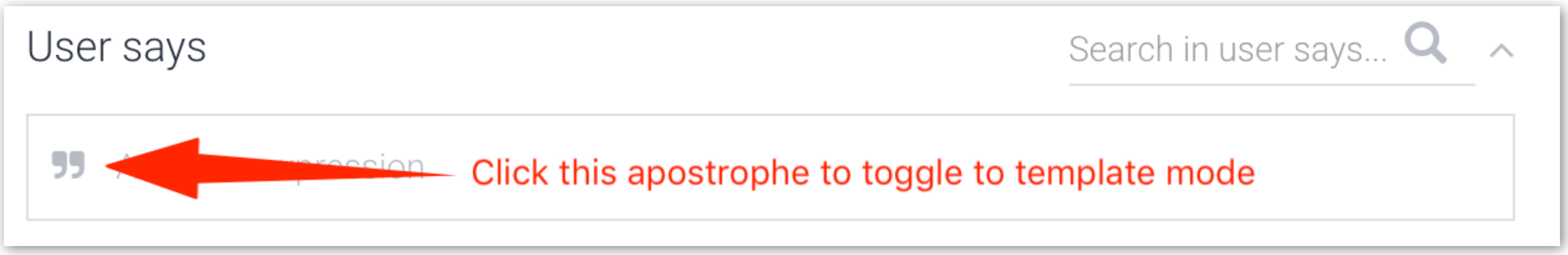


Figure 1.1 Click on the quotation mark to the left of the User says to toggle

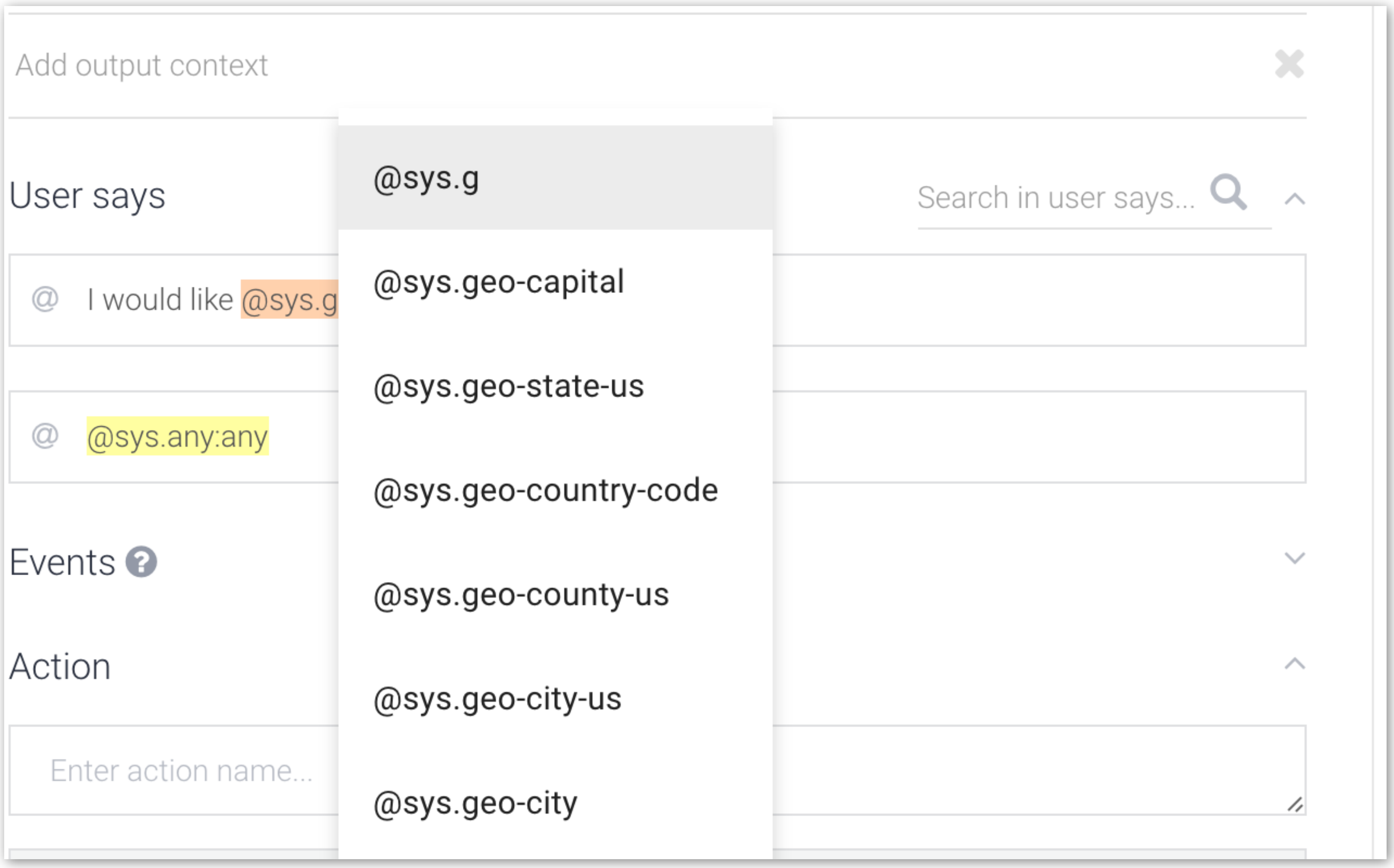


Figure 1.2 When you are in template mode, typing @ followed by a character should show you autocomplete options

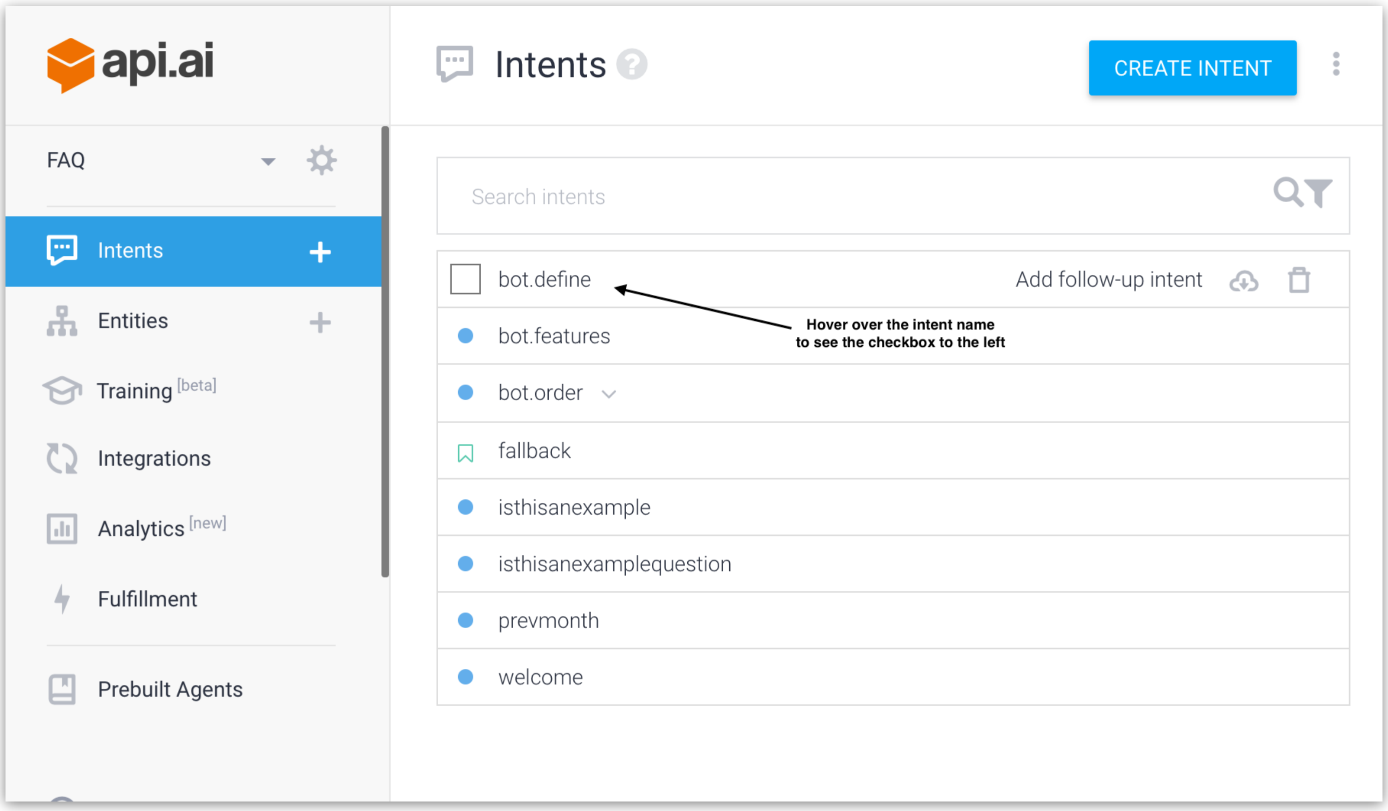
CHAPTER 2

SPEED UP THE PROCESS OF EDITING AGENTS

This feature is so hidden that many people don’t know about its existence.

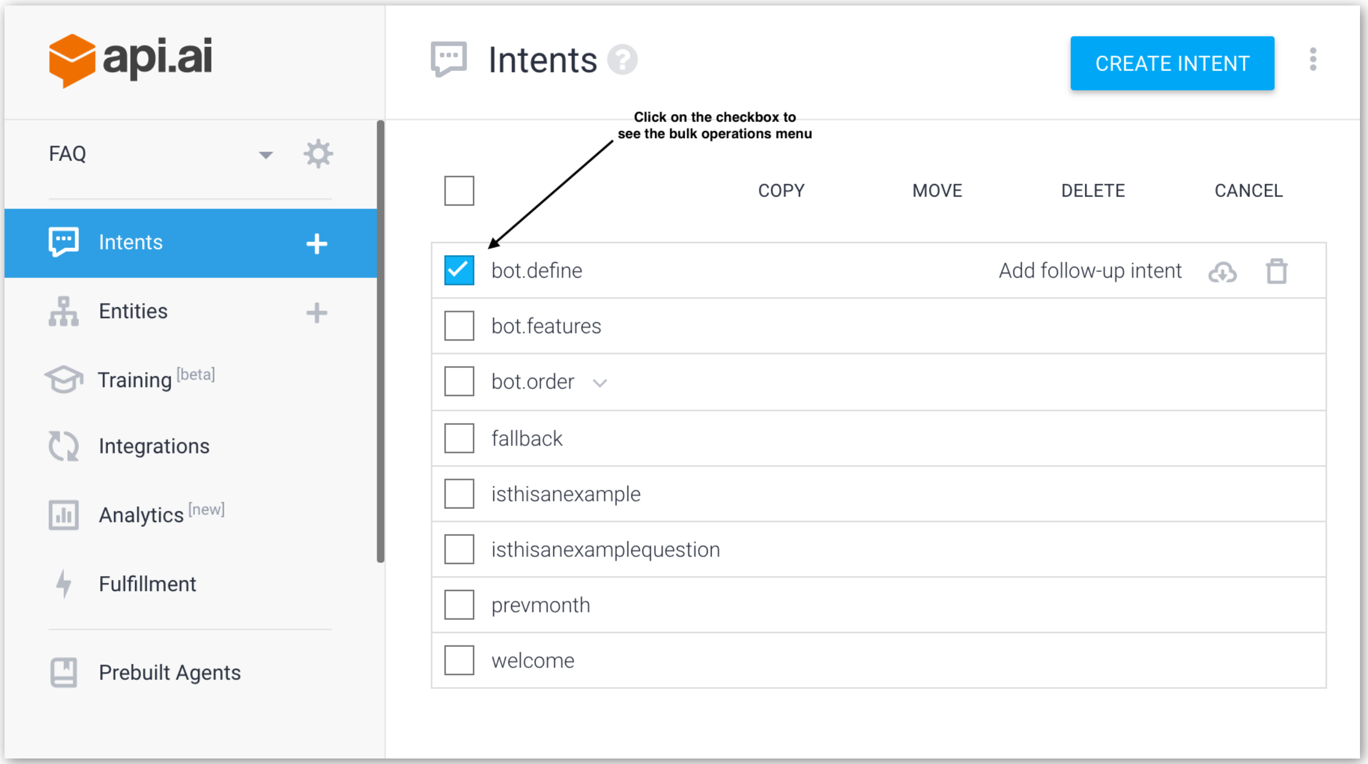
If you want to do bulk operations on entities and intents (e.g. you wish to delete all the intents without deleting the agent), you can make use of this feature. It is not one feature, but the common UI for doing bulk operations means we can pretend that it is a single feature.

To enable this, as you hover over the intent (or entity), you will notice a checkbox at the left of the intent. Once you click on the checkbox, the entire feature set of bulk operations becomes available to use. See the images to understand the sequence.



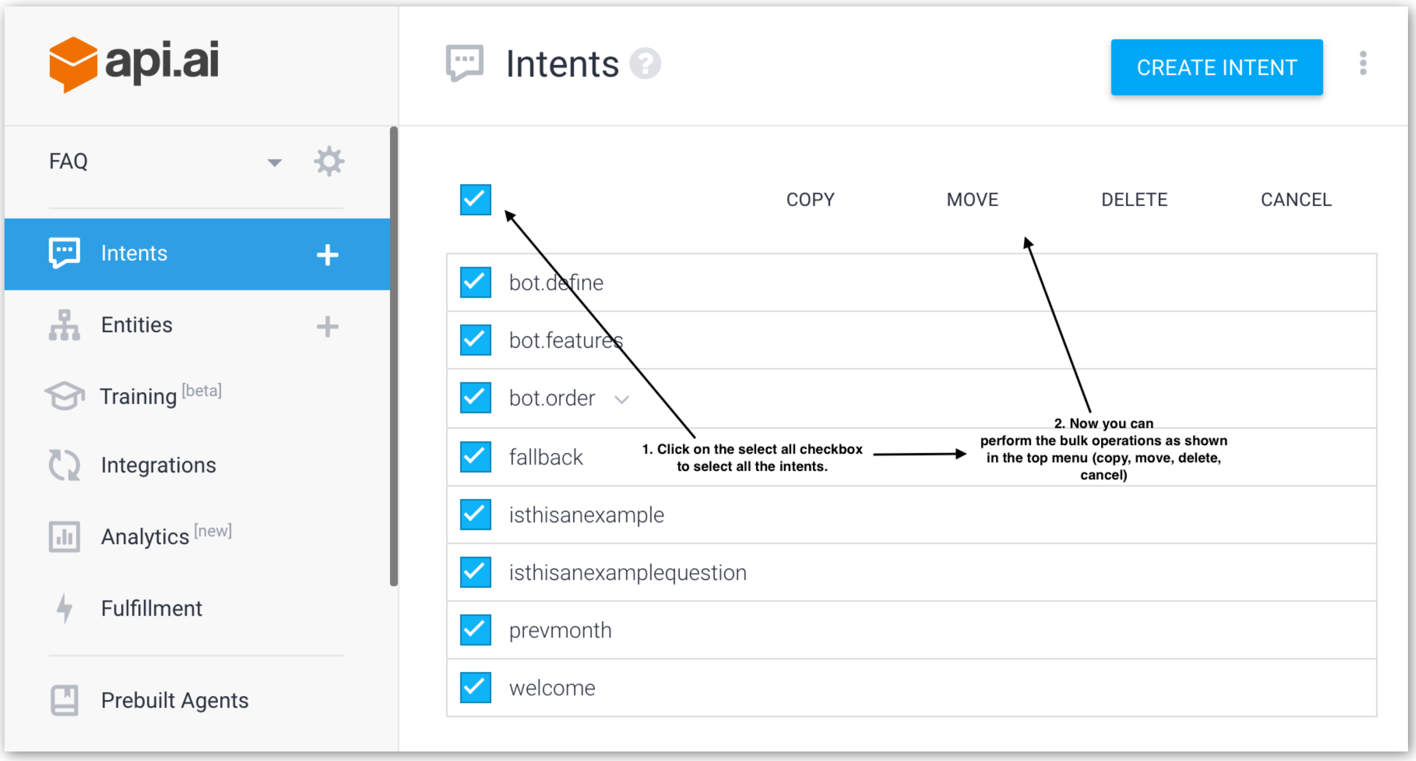
Hover over the intent name to see the checkbox to the left

Figure 2.1 Hover mouse over an intent to see the checkbox to the left



Click on the checkbox to see the bulk operations menu

Figure 2.2 Click on the checkbox. You will now see a checkbox at the top of the column as well as the list of possible bulk operations



1. Click on the select all checkbox to select all the intents.

2. Now you can perform the bulk operations as shown in the top menu (copy, move, delete, cancel)

Figure 2.3 Click on the top checkbox to select all the intents

CHAPTER 3

SPEED UP DATA ENTRY

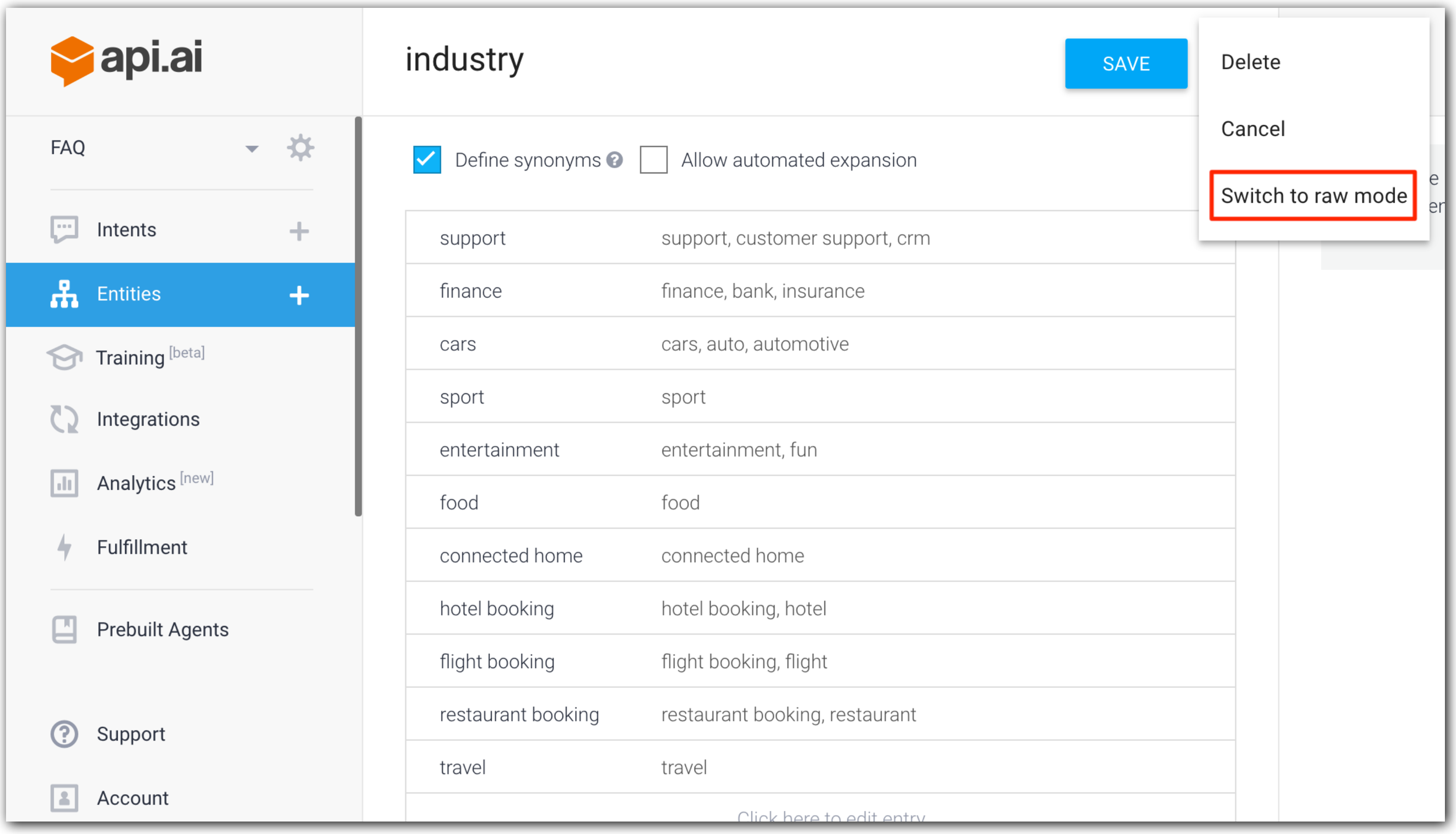
In API.AI, entities represent "concepts" - as they like to refer to them in the documentation.

In actual practice, we translate this to mean "object instances of a class". If you have been programming for a while, you are familiar with storing these objects in a database, and you also know that sometimes the number of entries in the table (i.e. the number of object instances) can be a substantial number.

While API.AI has an upper limit of 30000 on the number of values inside a given entity (as of this writing, June 2017), manually creating 30000 values is still a big task.

This is where you can use the CSV upload feature.

Click on the top menu inside the entity view (the top menu is the one with the three dots to the right of the entity's name). Click on "Switch to Raw Mode" if you are not in Raw Mode. Once you switch to Raw Mode, you will see a text area into which you can paste all your entity values in CSV format.



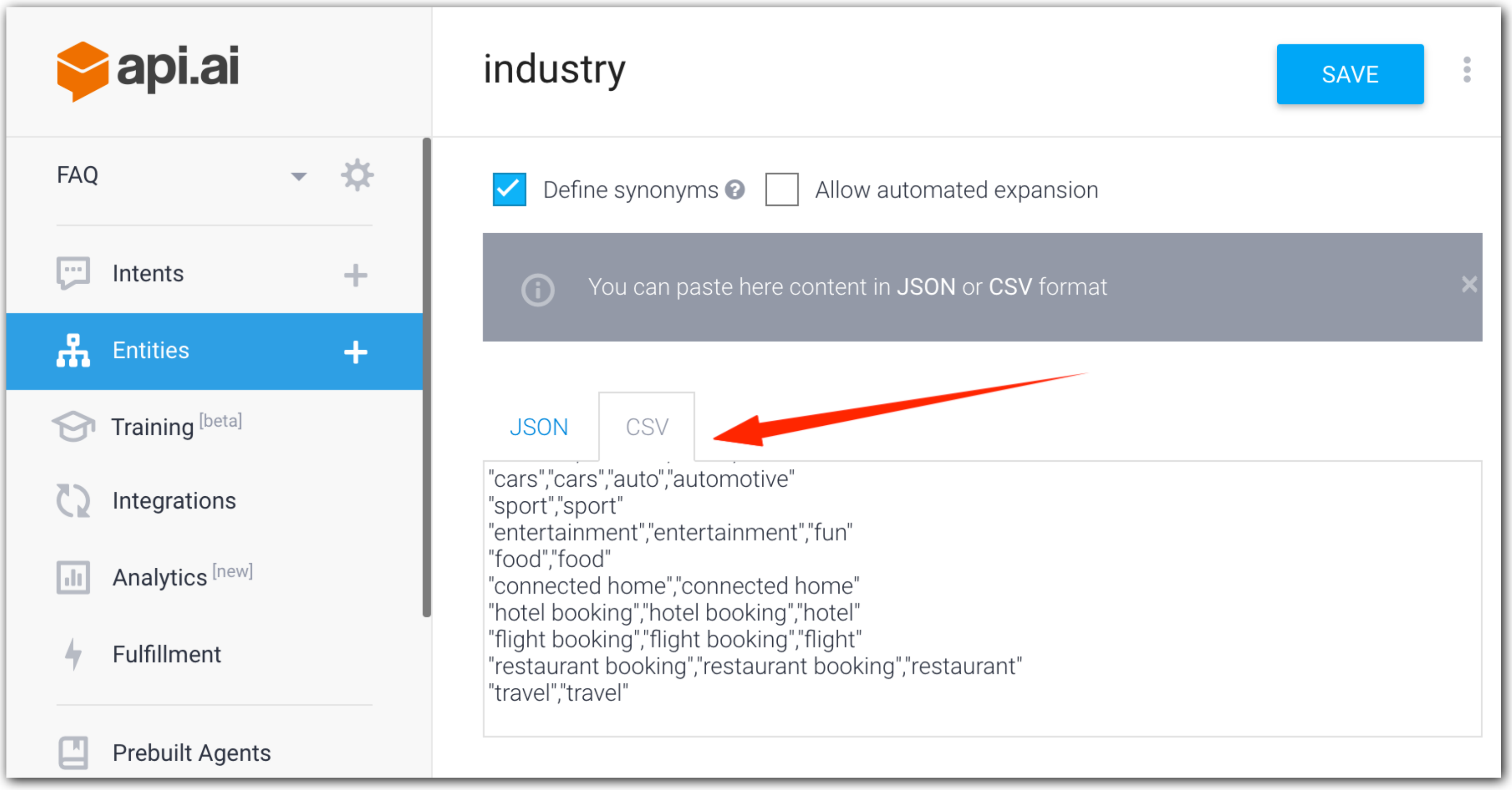
industry

Define synonyms ✓ Allow automated expansion ☐

support	support, customer support, crm
finance	finance, bank, insurance
cars	cars, auto, automotive
sport	sport
entertainment	entertainment, fun
food	food
connected home	connected home
hotel booking	hotel booking, hotel
flight booking	flight booking, flight
restaurant booking	restaurant booking, restaurant
travel	travel

Click here to edit entry

Figure 3.1 Click on Switch to raw mode to see the CSV input box



industry

Define synonyms ✓ Allow automated expansion ☐

You can paste here content in JSON or CSV format

```
"cars","cars","auto","automotive"
"sport","sport"
"entertainment","entertainment","fun"
"food","food"
"connected home","connected home"
"hotel booking","hotel booking","hotel"
"flight booking","flight booking","flight"
"restaurant booking","restaurant booking","restaurant"
"travel","travel"
```

Figure 3.2 Upload your entities using by copy/pasting your CSV data into the box above

Tip: You are most probably importing the values from either a spreadsheet, or a database. In most cases, you will usually have a feature to export the values out as CSV. There is a small catch though. Everything in the CSV file must be a string (that is, should be inside quotes), so you need to make sure you export the values correctly to the CSV format.

CHAPTER 4

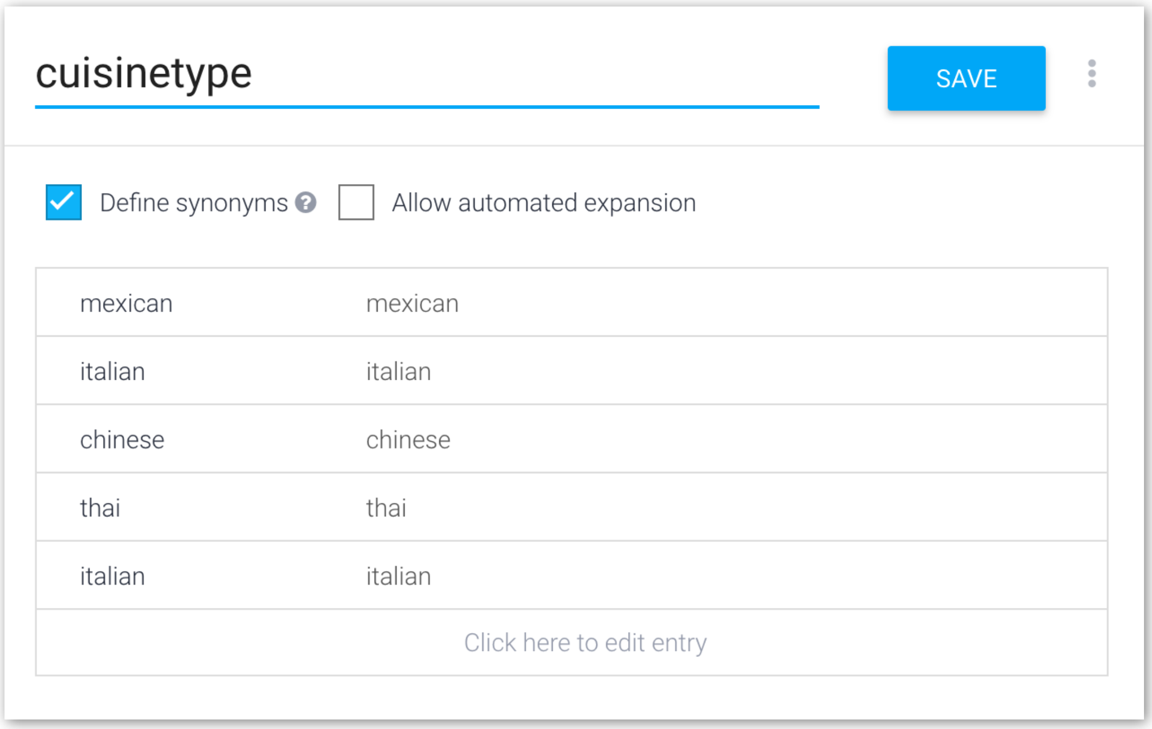
HANDLE COMPLEX DATA INPUT FROM USER

One of the nice features in API.AI is the “composite entity”.

When you are getting input from the user, you might have to handle complex data. As an example, say you wish to handle some kind of boolean logic in your API.AI intent. You want the user to be able to say “I want Mexican or Chinese food”, and capture the two types of cuisine. (Remember that the user could say any two cuisine types).

You could have a simple cuising type entity called @cuisinetype (Figure 4.1) which has values for all the different types of cuisines. Now you can “build” on top of this @cuisinetype and have an entity called @cuisinetypeor which you will define as shown in Figure 4.2.

In fact, you can even create a composite entity for “except”, where the user says “ I would like anything but Indian and Chinese”. (see Figure 4.3)

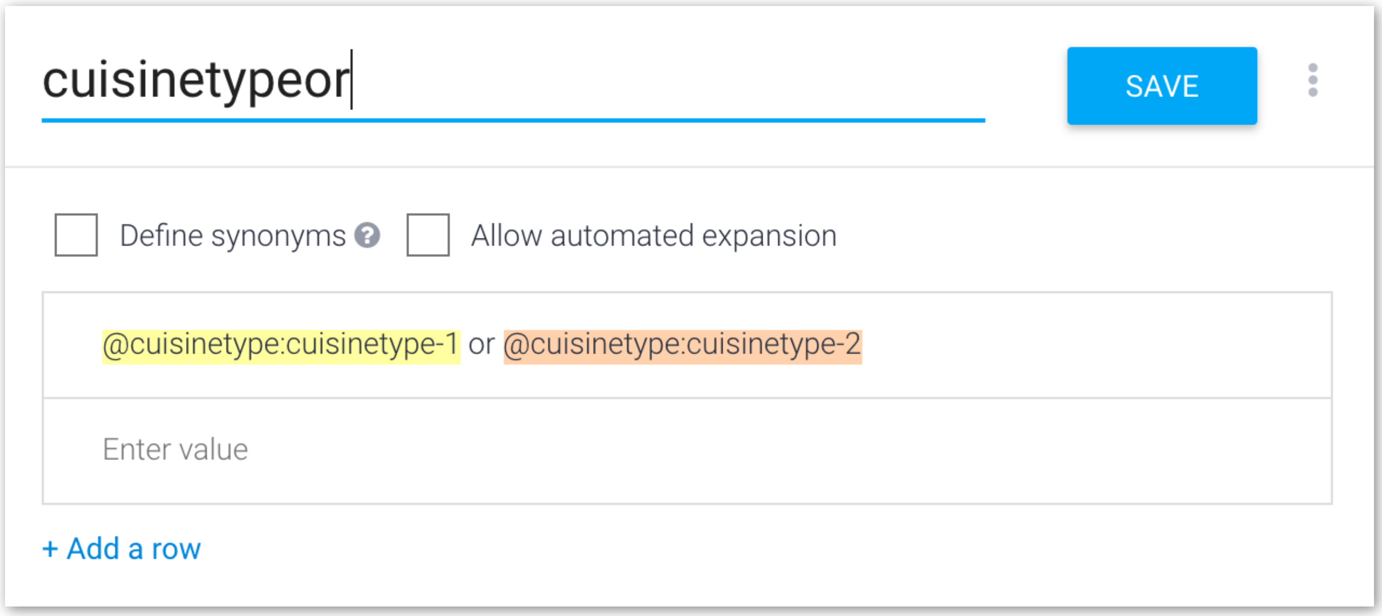


cuisinetype

[✓] Define synonyms [ ] Allow automated expansion

mexican	mexican
italian	italian
chinese	chinese
thai	thai
italian	italian

Figure 4.1 Declare the @cuisinetype entity

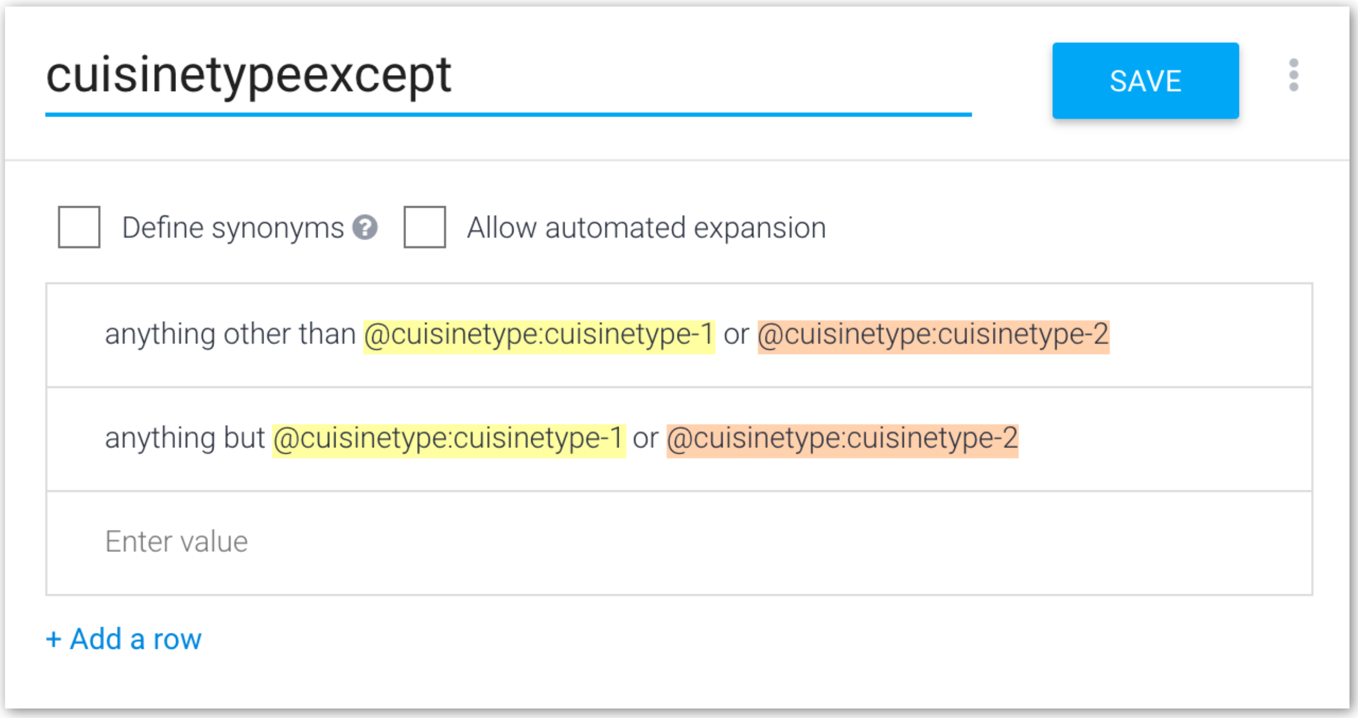


cuisinetypeor

[ ] Define synonyms [ ] Allow automated expansion

@cuisinetype:cuisinetype-1 or @cuisinetype:cuisinetype-2

Figure 4.2 Create @cuisinetypeor based on @cuisinetype. Remember to uncheck the Define synonyms checkbox



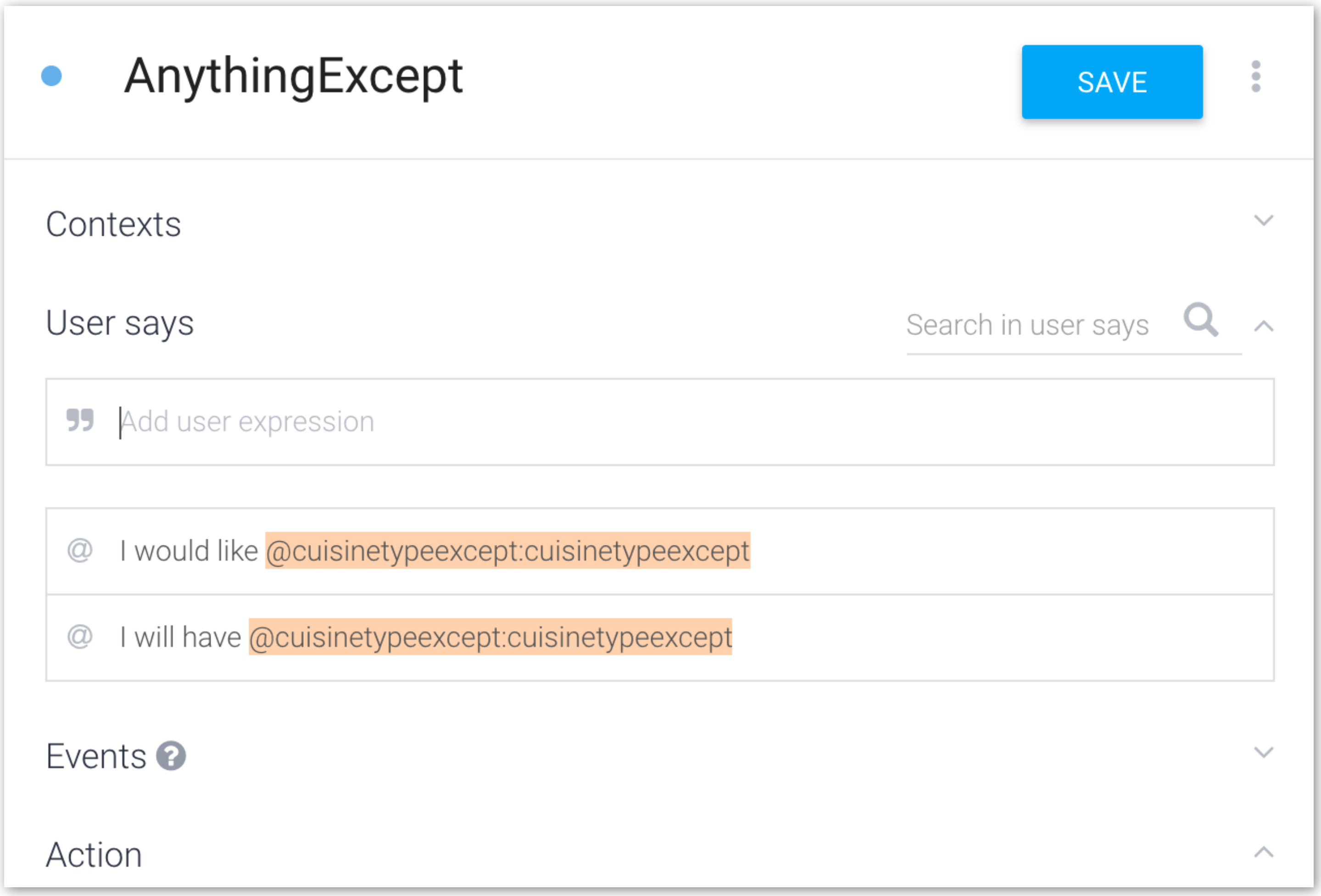
cuisinetypeexcept

[ ] Define synonyms [ ] Allow automated expansion

anything other than @cuisinetype:cuisinetype-1 or @cuisinetype:cuisinetype-2

anything but @cuisinetype:cuisinetype-1 or @cuisinetype:cuisinetype-2

Figure 4.3 Create the @cuisinetypeexcept entity


AnythingExcept

Contexts

User says

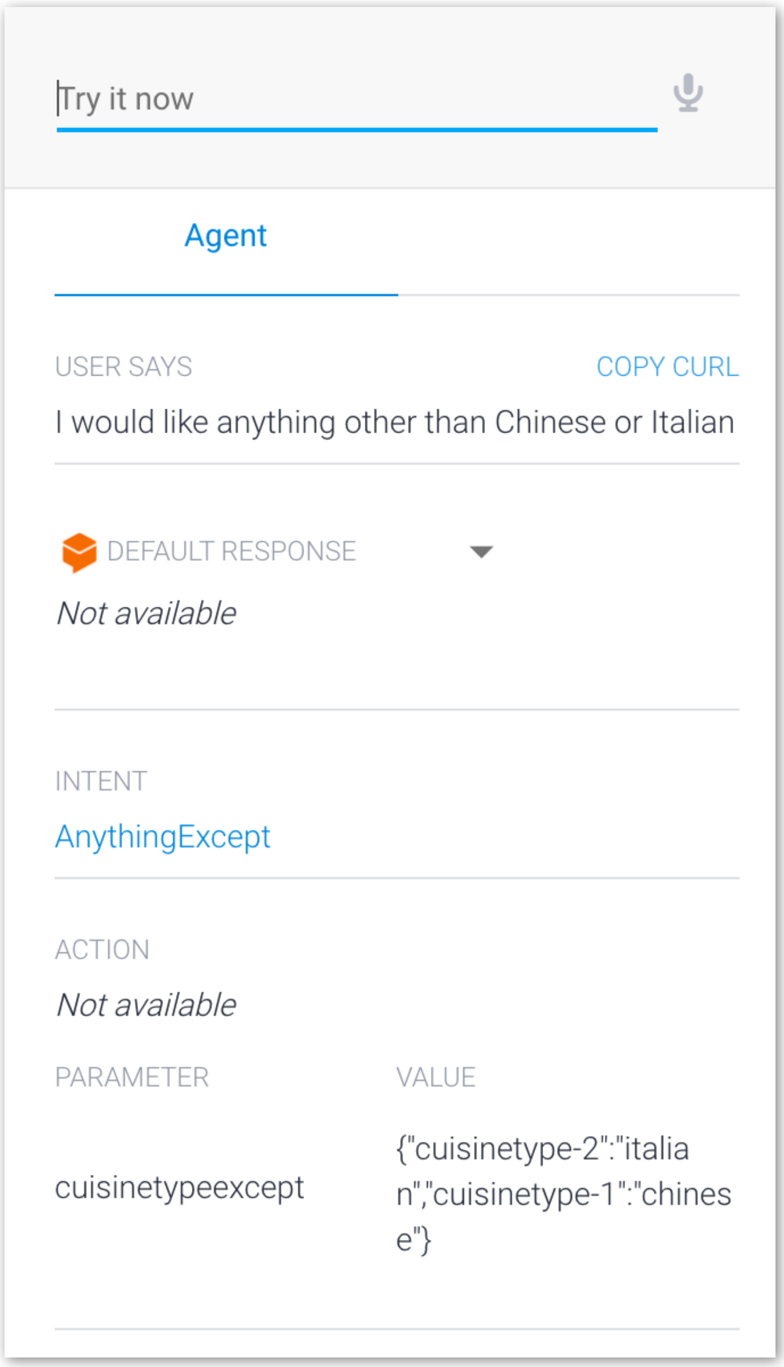
I would like @cuisinetypeexcept:cuisinetypeexcept

I will have @cuisinetypeexcept:cuisinetypeexcept

Events

Action

Figure 4.4 Declare the intent which uses @cuisinetypeexcept. Remember to add the user says expressions in template mode



USER SAYS

I would like anything other than Chinese or Italian

DEFAULT RESPONSE

Not available

INTENT

AnythingExcept

ACTION

Not available

PARAMETER	VALUE
cuisinetypeexcept	{"cuisinetype-2":"italian","cuisinetype-1":"chinese"}

Figure 4.5 The intent is correctly matched, AND the variables are correctly identified

The beauty of the composite entity is that you now have a single object to hold multiple values, making your code more structured and maintainable (especially as you pass these objects to your webhooks).

CHAPTER 5

HAVE THE AGENT INITIATE THE CONVERSATION

In API.AI, the agent is usually expected to “reply” to the user. In other words, the user has to initiate the conversation for the chatbot to get going.

You can see this even in the design of how intents are created. You first have the “user says” phrases up top, and then you have the “Text response” down at the bottom of an intent.

But what if you need to have the agent initiate the conversation? Sometimes users are a little tentative as to what they can expect from a chatbot, and may be reluctant to start the conversation. What do you do in those circumstances?

You will use the “Event” field in an intent. For example, if you inspect the Default Welcome intent of your chatbot, you will notice that the Event field has a string called WELCOME. Figure 5.1 shows the Default welcome intent of the florist chatbot.
OK, so how do we use it to initiate the conversation?

The REST API
For certain tasks, we *have* to use the REST API which is provided by API.AI. This is one of those. To start the conversation, we will make an API call to the REST API of your agent. To see more details on how to send this API call, you can read the following post. The important thing for us to understand is that you will “trigger” the default

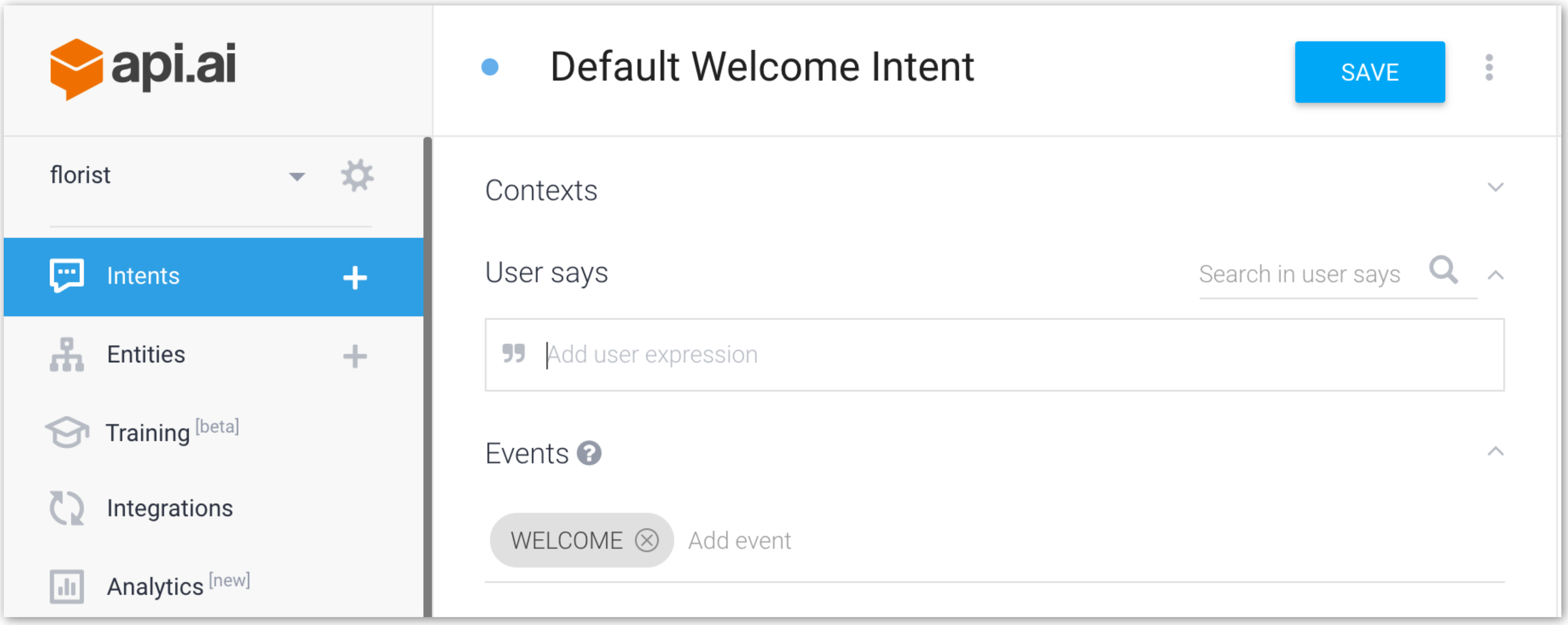
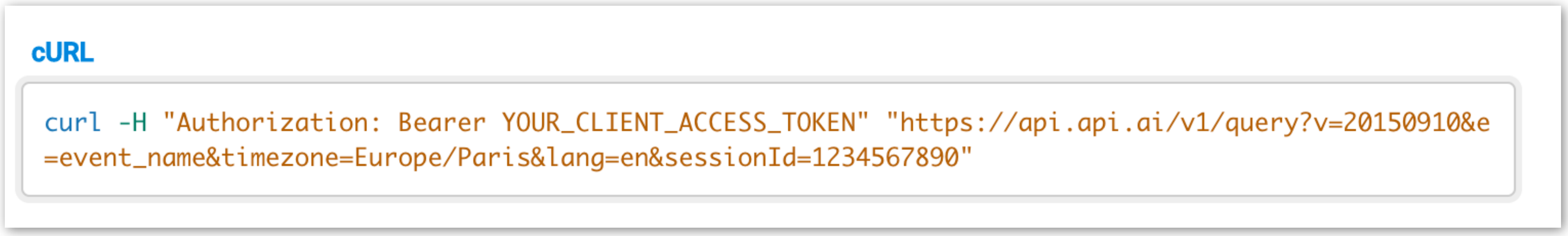


Figure 5.1 Notice the WELCOME string under the Events



cURL

```
curl -H "Authorization: Bearer YOUR_CLIENT_ACCESS_TOKEN" "https://api.api.ai/v1/query?v=20150910&e
=event_name&timezone=Europe/Paris&lang=en&sessionId=1234567890"
```

Figure 5.2 An example cURL request to trigger an event.
https://docs.api.ai/docs/concept-events#invoking-an-event-via-query-request

welcome intent programmatically by making sure that you pass an event parameter with the value of WELCOME.

So now that you know that the default welcome intent can be triggered by making a call to the REST API, it should follow that you can also do the following

1. You might want to send *two* responses in sequence to the user. The first response would be just what API.AI automatically does, but the second one could be triggered by using an Event.
2. Have your chatbot automatically poke a user if they don't respond within a certain period of time (so you don't lose the session)
3. Trigger a certain intent as a response to something which happens *outside* of the chat. For example, you can run asynchronous tasks in the background while a user is chatting and notify them once the task is complete by triggering the suitable event.

CHAPTER 6

COLLABORATE ON AGENT CREATION

The Share feature provided by API.AI is a simple and important one, but for some reason the API.AI team didn't make a big deal of the announcement (or at least that is what I think) and as a result many people don't seem to be aware of it. And its also all Google's fault!

OK, I am just kidding. But one of the reasons I *think* this feature is not more well known is because the top results when you search for "API.AI collaborate with a team" or similar all go to the different search results, none of which talk about the Share feature! Besides, the feature has only been out for about a month and a half, so maybe it will become more apparent in due time.

So here it is in a nutshell: just go to this link and you will learn all there is to know about it (which isn't much).
https://docs.api.ai/docs/concept-agents#share

Once you read that link, also check out my blog post about the feature because it covers a couple of minor nitpicks in the way the Share feature is currently implemented, so you know what's coming.
http://miningbusinessdata.com/api-ai-team-account/

CHAPTER 7

START FROM A SMART AGENT

If you are wondering if it is possible to create a “smart” chatbot without too much effort, here is a suggestion: Make use of the prebuilt SmallTalk agent.

While you might know that API.AI has this thing called the SmallTalk domain which you can turn on and off, using the SmallTalk domain has a few problems:

1. You don’t really know all the questions it can answer, because it is a black box
2. You can’t modify or control the answers it can provide for the most part
3. The Smalltalk domain intents sometimes can interfere with similar intents that you the user defined, and it is an all-or-nothing proposition because you cannot selectively turn off only those intents.

To get around this, the API.AI team released the capabilities of the SmallTalk agent as a Prebuilt agent. (Don’t know what prebuilt agents are? Check out this link).

So why are prebuilt agents better than enabling the domain? It is because you can modify each and every intent in the prebuilt agent, so it is completely transparent. Does an intent in the prebuilt agent conflict with your intent? Just delete the conflicting one in the prebuilt agent!

Here is how I recommend you start with a smart agent.

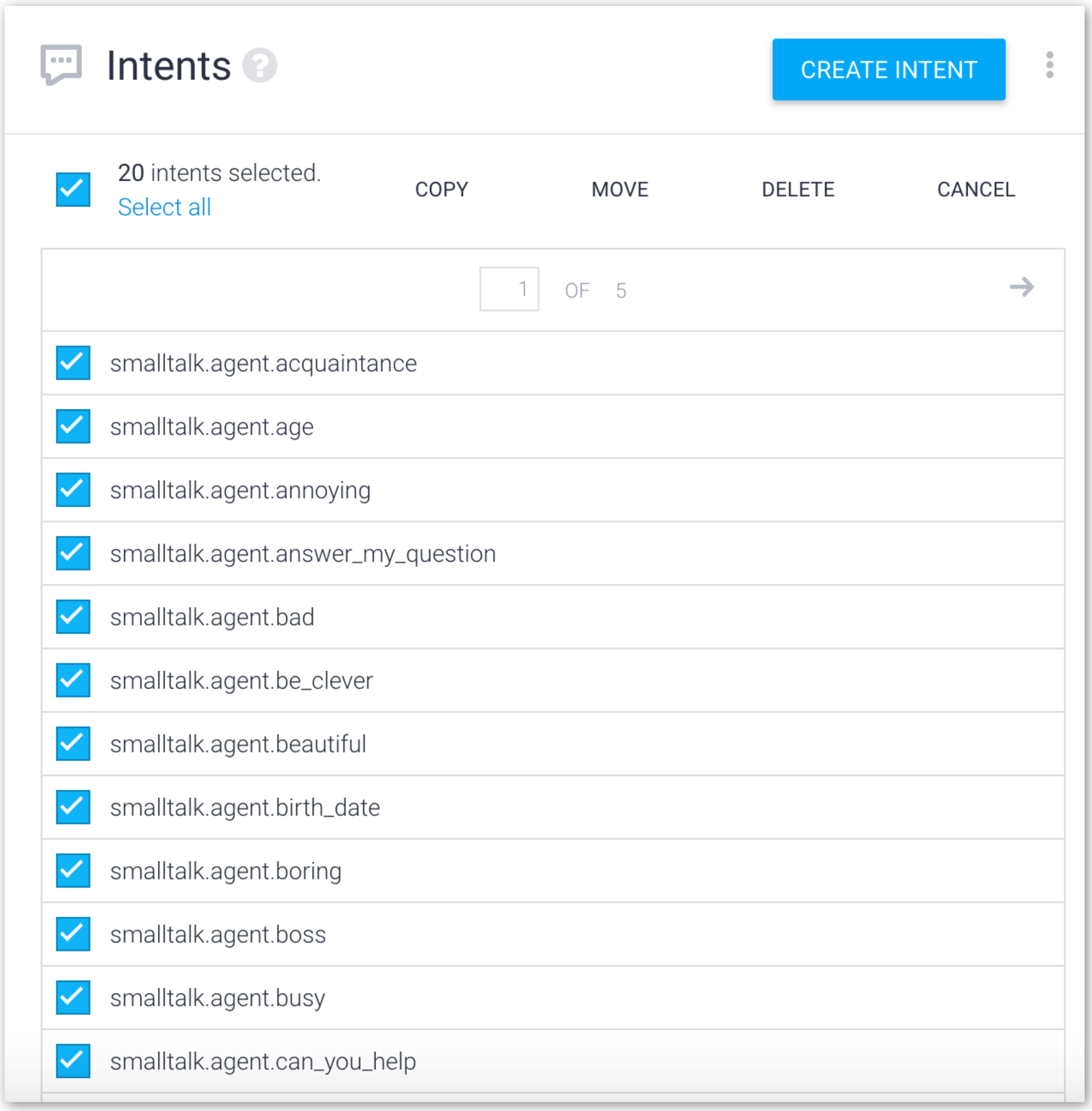
1. Once you create a new agent, go to the Prebuilt agents tab and select the SmallTalk agent

2. Import the SmallTalk agent into your new agent
3. Add all your custom intents after you do this import

Have you already created an agent? You can do the following:

1. Import the prebuilt SmallTalk agent into a new agent
2. Bulk select the intents from the SmallTalk agent (use the steps described in Chapter 2)
3. Copy all the intents over into your existing agent

See figures 7.1 and 7.2 for more details.


Intents

CREATE INTENT

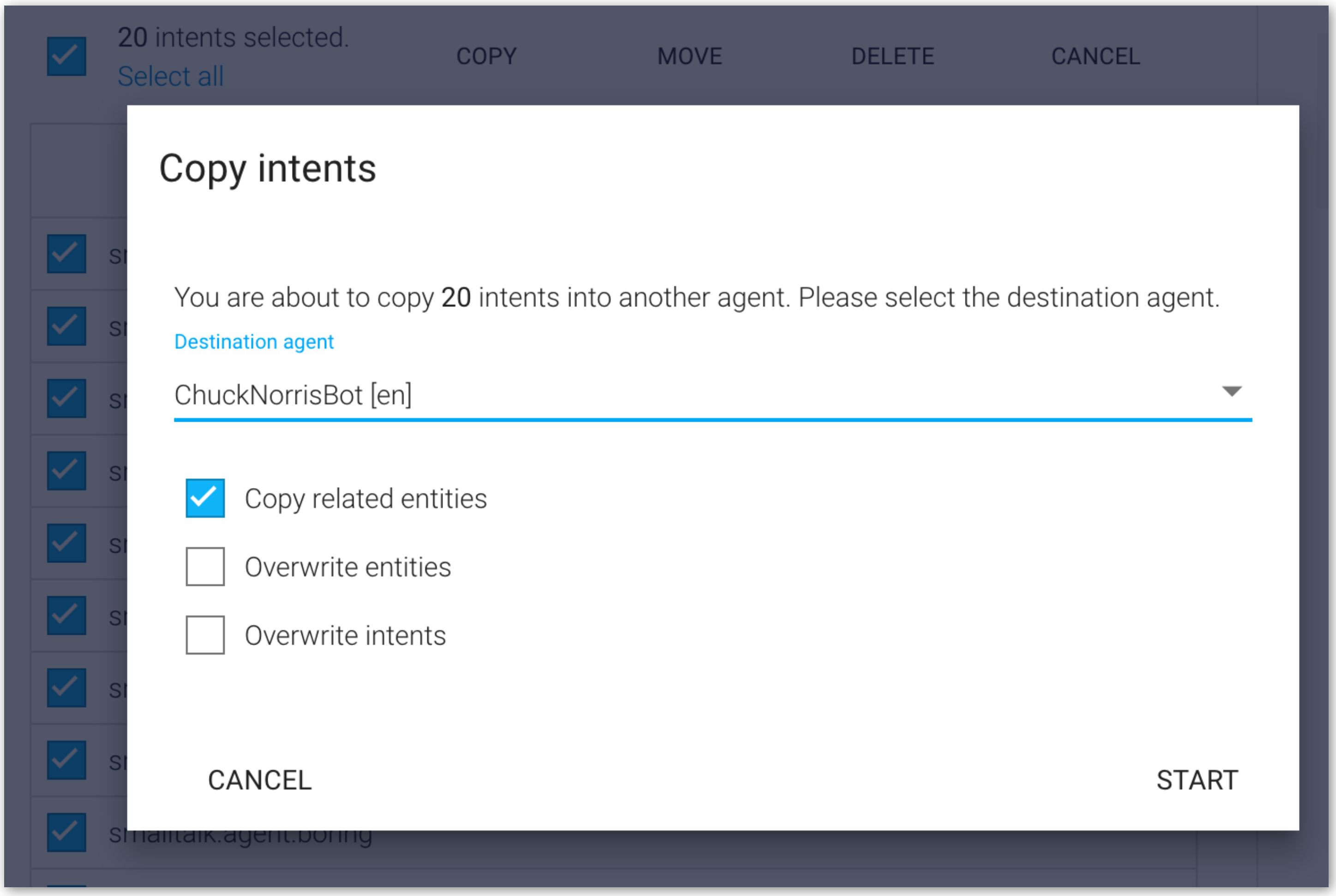
20 intents selected.
Select all

COPY MOVE DELETE CANCEL

1 OF 5

- ✓ smalltalk.agent.acquaintance
- ✓ smalltalk.agent.age
- ✓ smalltalk.agent.annoying
- ✓ smalltalk.agent.answer_my_question
- ✓ smalltalk.agent.bad
- ✓ smalltalk.agent.be_clever
- ✓ smalltalk.agent.beautiful
- ✓ smalltalk.agent.birth_date
- ✓ smalltalk.agent.boring
- ✓ smalltalk.agent.boss
- ✓ smalltalk.agent.busy
- ✓ smalltalk.agent.can_you_help

Figure 7.1 Bulk copy all intents from your smalltalk agent


20 intents selected.
Select all

COPY MOVE DELETE CANCEL

Copy intents

You are about to copy **20** intents into another agent. Please select the destination agent.

Destination agent

ChuckNorrisBot [en]

- [✓] Copy related entities
- [ ] Overwrite entities
- [ ] Overwrite intents

CANCEL START

Figure 7.2 Copy/paste the SmallTalk intents into your existing agent

CHAPTER 8

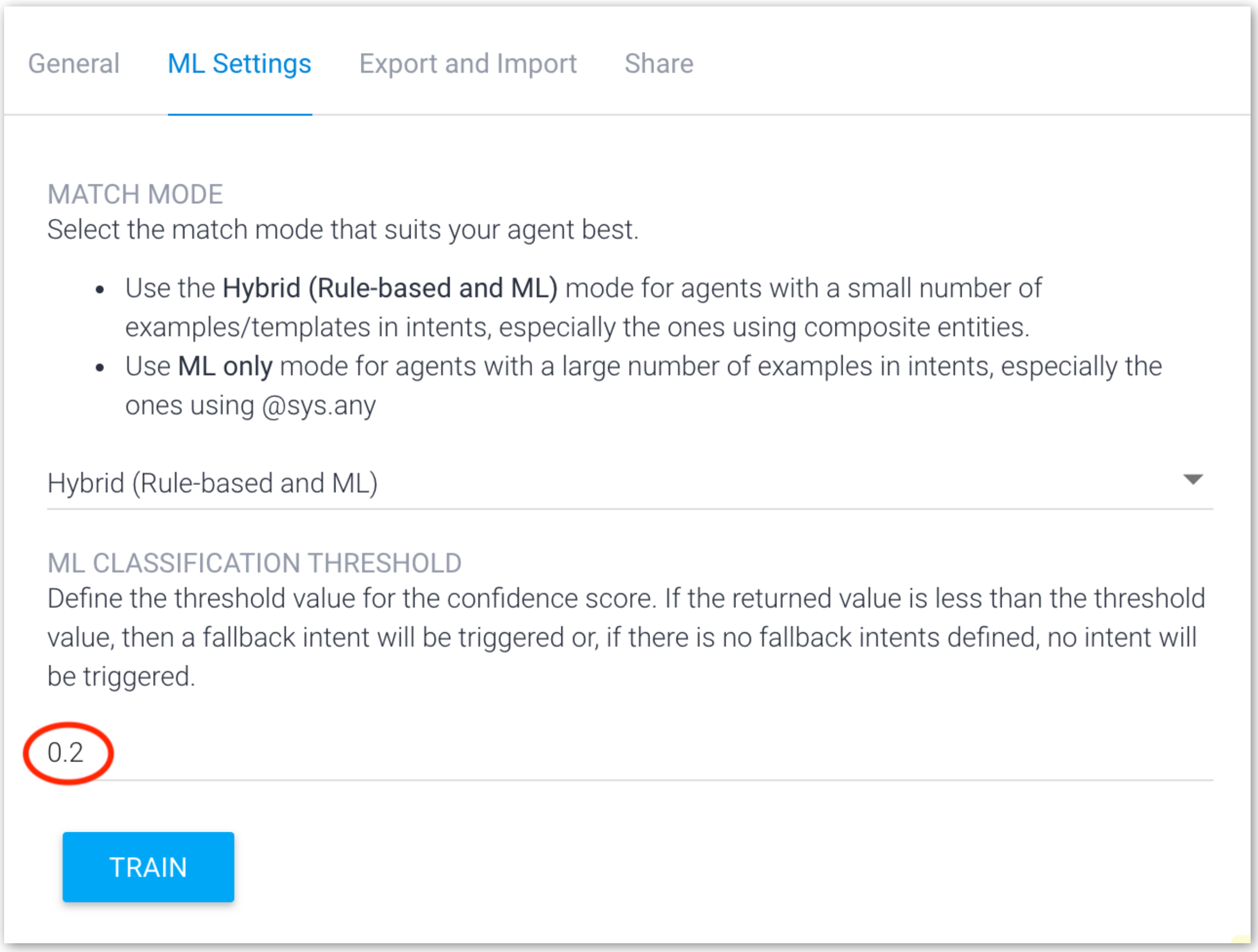
TUNE THE MATCHING ACCURACY

One of the frequent issues raised on the API.AI discussion forum is how no matter what the user says, the Default Fallback intent is always triggered! Usually, this is a sign that the Machine Learning is trying to match the intent to the user says phrases a *little too strictly*.

And there is a way to fix this problem. Under the agent settings (click on the gear like icon to the right of the agent's name), you will have a tab called ML settings (see Figure 8.1).

The ML threshold can be increased or decreased based on how the agent is performing. Here is a rule of thumb: if the Default Fallback intent is always triggered as mentioned before, then decrease the ML threshold. On the other hand, if the Default Fallback is never triggered no matter what, then increase the ML threshold. Use the Default Fallback like a compass to help guide you towards an appropriate level of ML matching for your agent.

A small caveat: the ML threshold cannot be set on an intent-to-intent basis. This means the same threshold will apply to all your intents.



MATCH MODE
Select the match mode that suits your agent best.

- Use the **Hybrid (Rule-based and ML)** mode for agents with a small number of examples/templates in intents, especially the ones using composite entities.
- Use **ML only** mode for agents with a large number of examples in intents, especially the ones using @sys.any

Hybrid (Rule-based and ML)

ML CLASSIFICATION THRESHOLD
Define the threshold value for the confidence score. If the returned value is less than the threshold value, then a fallback intent will be triggered or, if there is no fallback intents defined, no intent will be triggered.

0.2

Figure 8.1 ML classification threshold can be used to tune the matching accuracy

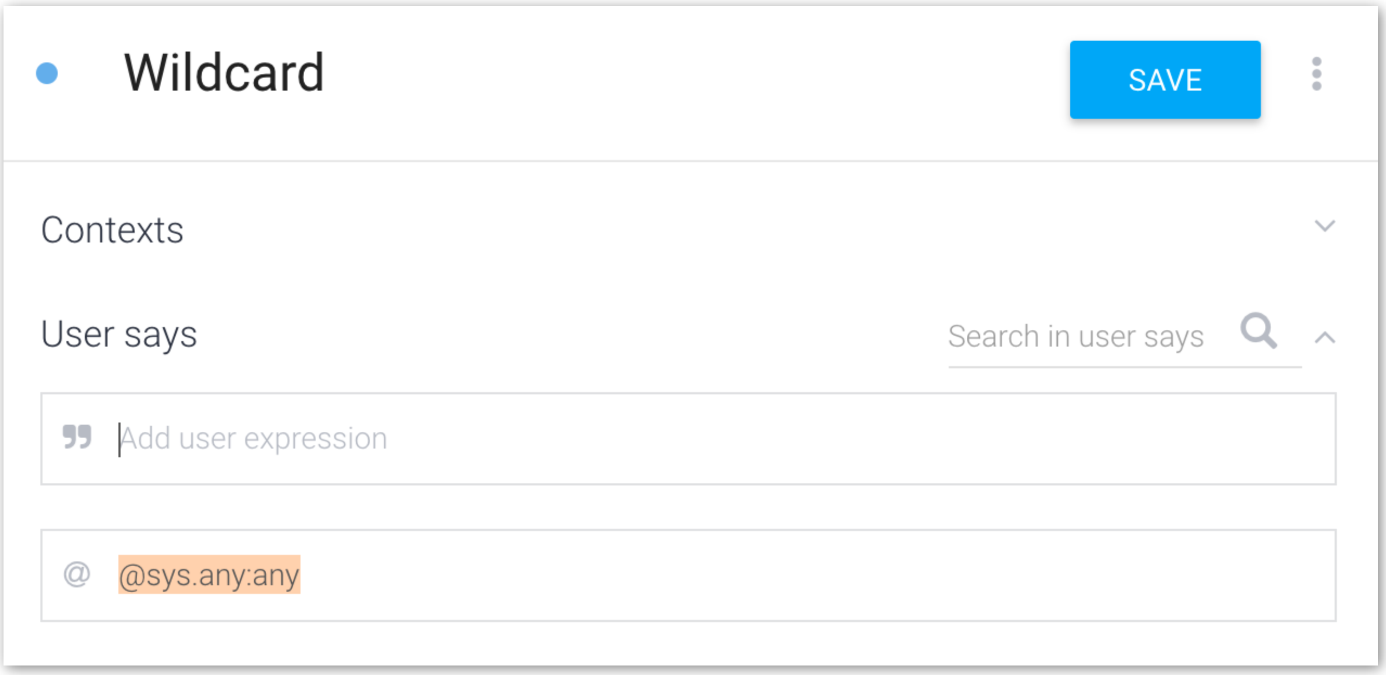
CHAPTER 9

INCREASE ERROR TOLERANCE TO USER INPUT

One of the absolutely certain things with creating chatbots is that you can never be certain what a given user will say as a response to your question.

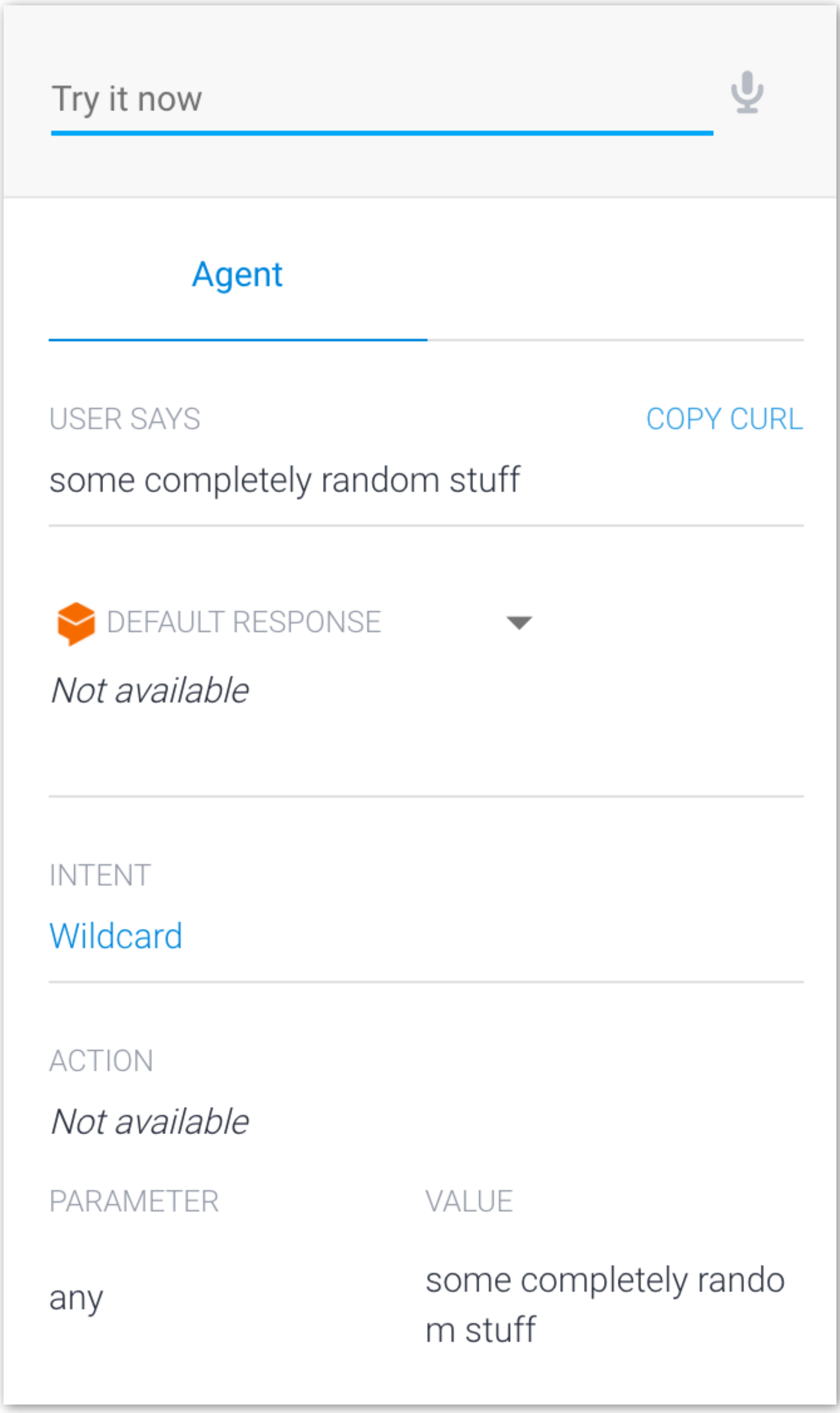
Suppose you want to handle the user's response no matter what they say. Do you write out 10,000 intents to cover every possible scenario? Of course not. You pull out the @sys.any wildcard.

So you would declare a user says in template mode and the only thing you will write out is @sys.any. That's it. (see Figure 9.1 and 9.2). Now, no matter what the user says, it will capture this input. From the testing I have done, it does seem like using the @sys.any wildcard in this way does not interfere with other intents (in other words, API.AI doesn't map every utterance to the wildcard intent), but it is a good idea to use this @sys.any approach only in intents which already have input contexts.



@sys.any:any

Figure 9.1 In a pinch, you can resort to the @sys.any wildcard which will match ANY user input



some completely random stuff

Not available

INTENT
Wildcard

ACTION
Not available

PARAMETER	VALUE
any	some completely random stuff

Figure 9.2 Console output for the wildcard intent above

CHAPTER 10

MAKE YOUR AGENT SMARTER OVER TIME

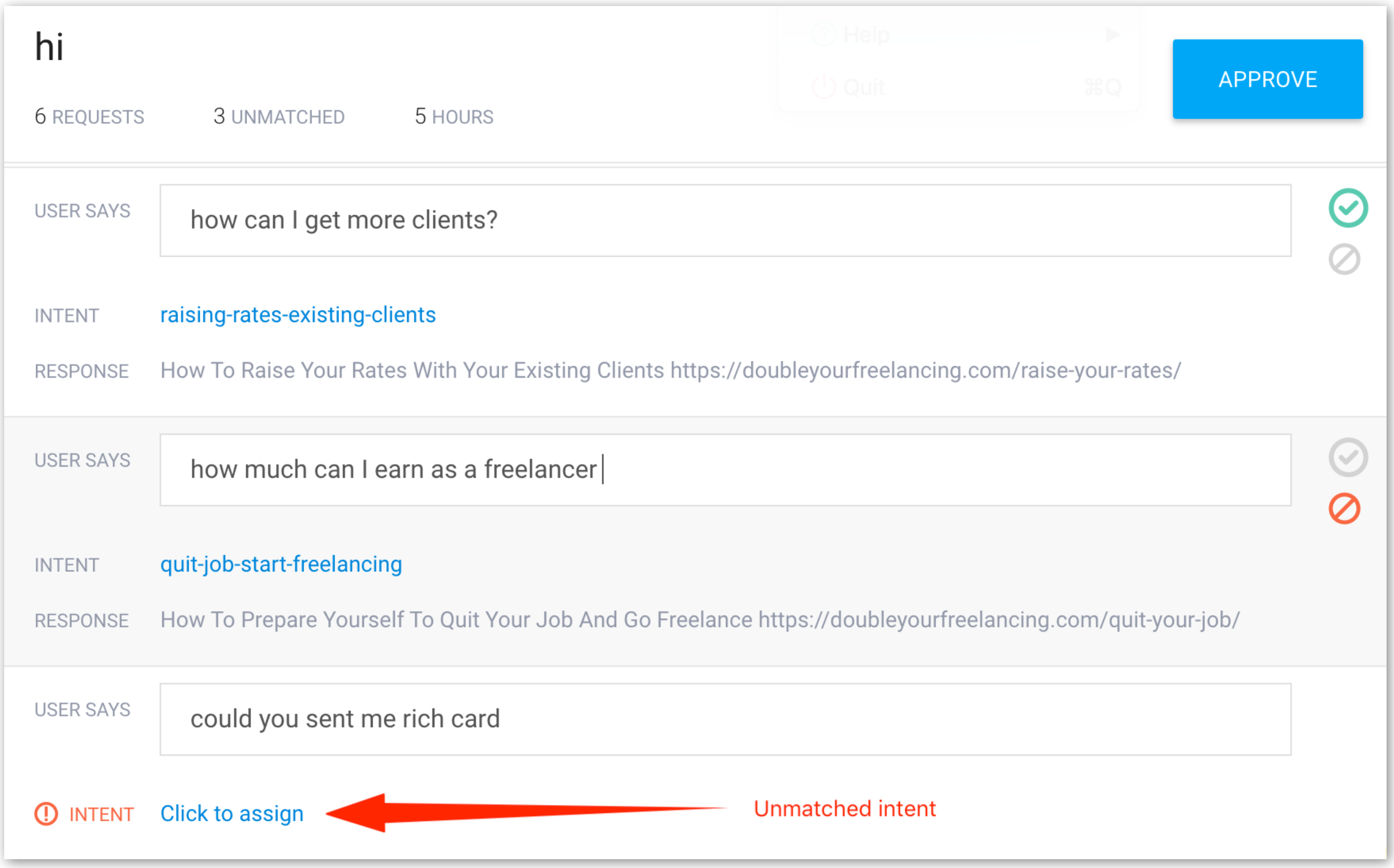
Did you know that you can make your API.AI agent smarter over time?

Once your agent starts fielding queries and providing responses, you can inspect the Training/History tab. Here, you have the option to do the following:

1. If the agent should have responded to a certain query but didn't, then select the correct intent to match
2. If the agent did respond to a query by matching it to the wrong intent, then you can ask the agent to ignore the phrase (i.e. don't match it to the intent).

Once you are done, you will click on Approve to save your changes (see Figure 10.1 for an example). After you approve the changes, your agent will be trained in a few minutes with your new information. In other words, the agent is actually getting smarter over time in how it responds to user requests.

Tip: If you want your chatbot to be as accurate as possible, you should set aside some time each week to look at the Training section and help improve your bot. Also consider creating new intents when you see a certain class of questions being repeatedly asked and going to the Default Fallback intent. (cheap customer survey/research!)



hi

6 REQUESTS 3 UNMATCHED 5 HOURS

APPROVE

USER SAYS how can I get more clients?

INTENT raising-rates-existing-clients

RESPONSE How To Raise Your Rates With Your Existing Clients https://doubleyourfreelancing.com/raise-your-rates/

USER SAYS how much can I earn as a freelancer

INTENT quit-job-start-freelancing

RESPONSE How To Prepare Yourself To Quit Your Job And Go Freelance https://doubleyourfreelancing.com/quit-your-job/

USER SAYS could you sent me rich card

INTENT Click to assign

Figure 10.1 The training tab. You can agree with the assigned intent (green checkbox circle), disagree by asking API.AI to ignore the phrase (red no circle), and also assign unmatched intents. Click the Approve button at the top once you are done to complete the training.

BONUS CHAPTER

TIPS FOR WRITING COPY FOR YOUR CHATBOT

Recently, Google has been releasing some documentation around designing conversations for chatbot. Apparently, conversation design is something you need to keep in mind while creating your chatbot.

Why does conversation design matter?
For one thing, if you follow the principles laid out in their documentation on designing conversations, your chatbot is more likely to be accepted for Google Assistant. In addition, following conversation design principles will simply make your chatbot easier to chat with and also more conversational.
Since your users will engage with your chatbot more if it is more conversational, it would actually be a good idea to start thinking about conversation design even as you start creating your chatbot. Think of it this way: you are probably going to wrack your brain to come up with good responses in your chatbot anyway, so why not also follow some guidelines in the process? :-)

Tips for writing good copy
While the topic of conversation design can fill out an entire book by itself, here are three tips to help you write better copy for your chatbot.

1. Don't be too literal
For example, instead of using a literal instruction like this one:

Editing your shopping list is easy. To add an item, just say "Add," followed by the item you'd like to add. To remove an item, just say "Remove," followed by the item you'd like to remove.

Consider being intuitively obvious instead:
By the way, to edit your list, you can say something like "Add toothpaste" or "Take off the ice cream."

2. Communicate what was understood

User: Who made the statue David?
UI: David was created by Michelangelo.

3. Give users credit and save the guidance for those who need it

Avoid this:

User: Start a metronome.
UI: What tempo did you want to start with? You can say, for instance, "110 beats per minute." Or you can give me a tempo like "Allegro" or "Moderately fast." (We can always speed it up or slow it down later.)
User: 92 beats per minute.
UI: Okay, 92 beats per minute. Here you go.

This is bad behavior on the part of the UI. It is asking a question and then immediately continuing to talk without handing over the turn to the user. This approach also requires the person either to wait for the lengthy initial message to finish or to interrupt, also causing them to be a poor conversation participant.

Instead, consider the sequential, time-consuming nature of conversation, and yield the speaking turn back to the user:

User: Start a metronome.
UI: Sure, what tempo?
User: [No reply]

UI: You can say, for instance, "110 beats per minute." Or you can give me a tempo like "Allegro" or "Moderately fast." (We can always speed it up or slow it down later.)
User: 92 beats per minute.
UI: Okay, 92 beats per minute. Here you go.

I am putting together a conversation design toolkit on my blog, based on tips provided by Google as well as material from other sources I am reading. If you want to learn more, go to my website and select “Conversation Design” from the Resources menu.

CHAPTER 12

LEARN MORE

To learn more about API.AI in particular and chatbots in general, visit http://www.miningbusinessdata.com

If you notice any mistakes and errors, please email me at aravind.mohanoor@gmail.com.

If there is any errata then I will be adding it as a supplement to this eBook in the same Gumroad product page (you will be notified via email). So do be on the lookout for those.

Happy learning!
Aravind Mohanoor
June 2017