Exercise 1: Create, modify, and delete entities

Entities recognize and capture specific pieces of information in the user input. In our flower shop chain chatbot, people asking us about store hours and locations might provide a specific location.

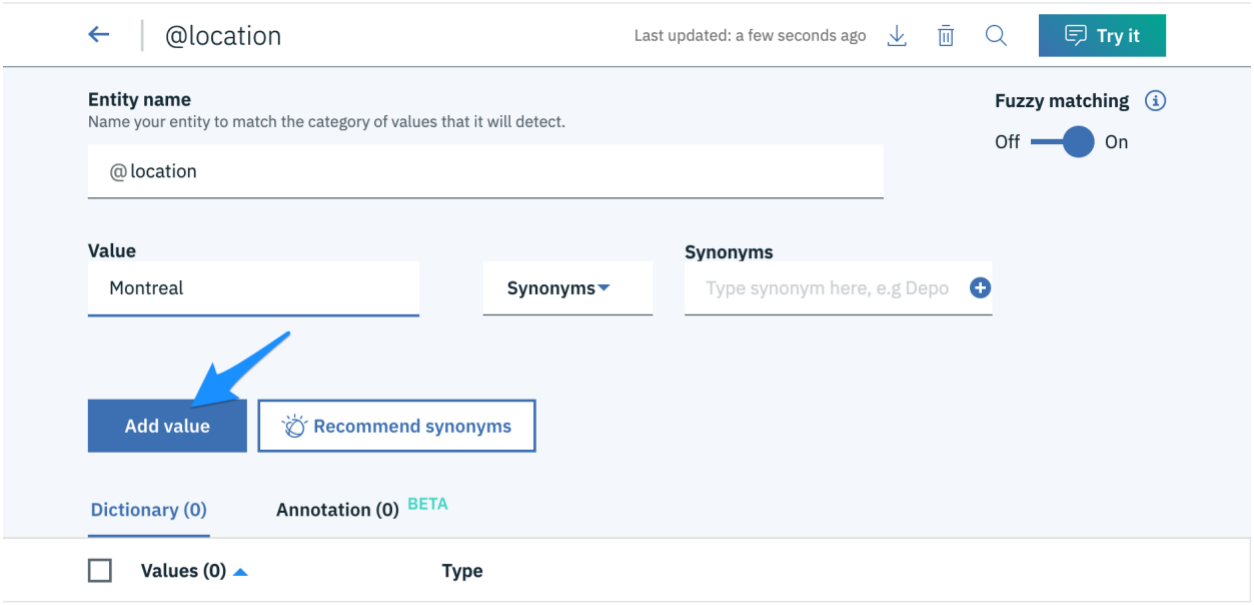
In our fictitious Flower Shop chain, we have stores in Toronto, Montreal, Calgary, and Vancouver. So, when a user asks, `Where is your Toronto store?` we shouldn't ignore that extra bit of information so that we can take the location into account when formulating a response.

We can start by creating a `@location` entity for those cities.

1. **In your skill, click on *Entities*** to enter the entities section.

2. Here, **click the *Create entity* button**. Choose `@location` as the entity name (note that the @ symbol is automatically added for you). Leave *Fuzzy Matching* enabled so that we can still detect the city name even if the user misspells it. Finally, **click the *Create entity* button**.

3. You'll be prompted to enter entity values and possible synonyms. **Enter `Montreal` and then click *Add Value*** to add this entity value to our entity.

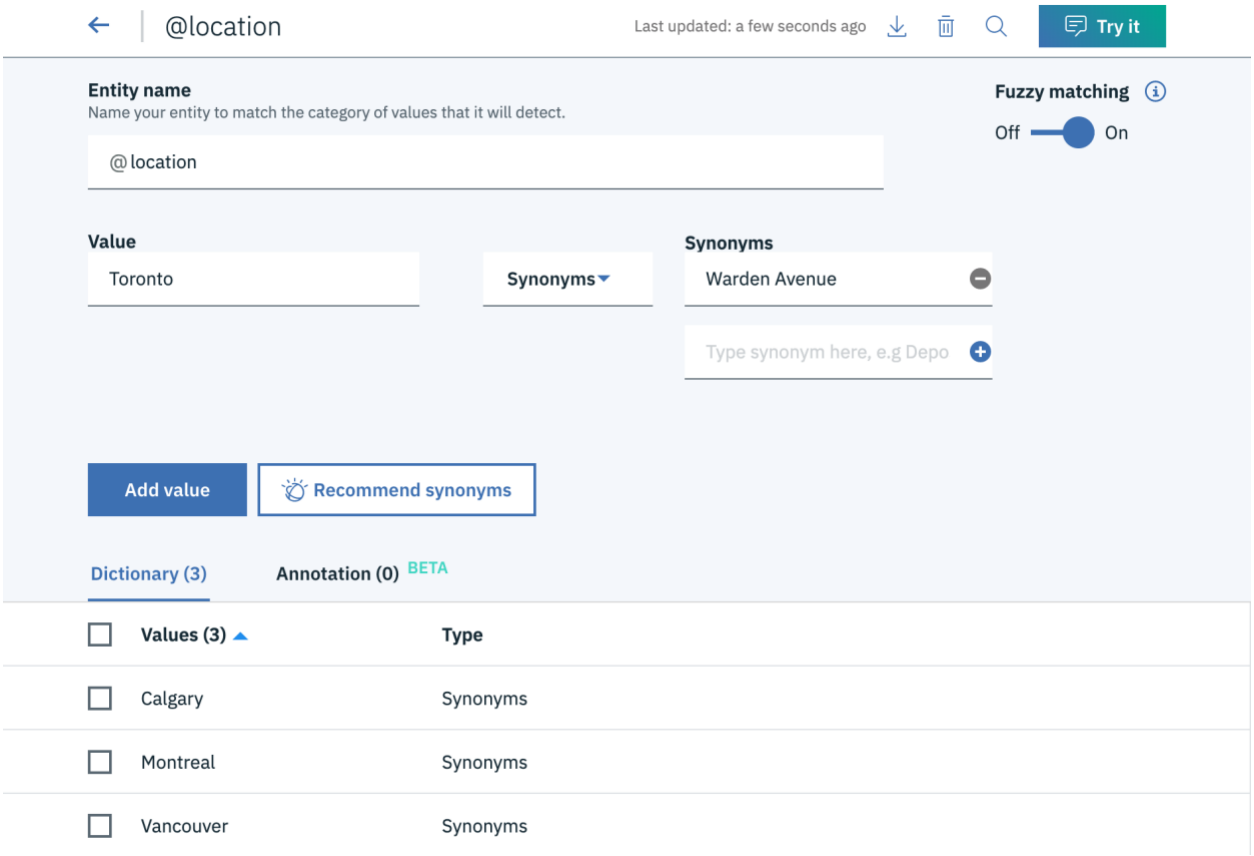


Generally speaking, you won't need a synonym for cities, but you might include some if the city has common nicknames or if people refer to your store location by its street or neighbourhood in the city. Nearby small cities and towns can also act as synonyms. After all, if people are

asking about your store in a nearby town, they might be happy with an answer for the nearest city.

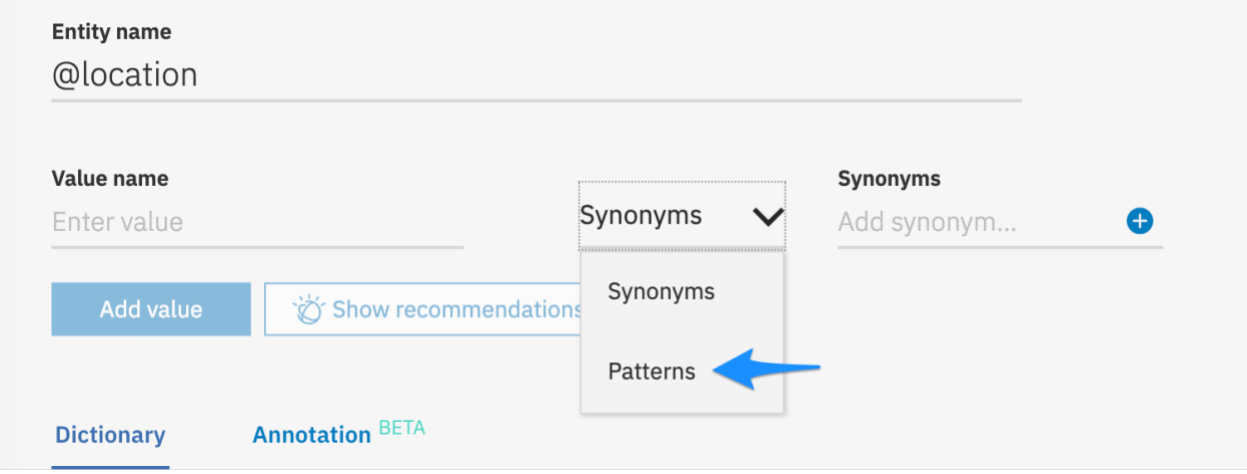
Essentially, a synonym is not necessarily the dictionary definition of synonym. Though those are good candidates for synonyms as well when it makes sense. For example, we could have an entity called `@relationship` and the entity value `@relationship:mother` with `mom` as a synonym for that value. When the user enters a question including the word `mom`, Watson will detect `@relationship:mother` (the entity value for that synonym).

4. **Repeat the process** for `Calgary` and `Vancouver`. Next, add `Toronto` as well. But for `Toronto`, add `Warden Avenue` as a synonym, as shown in the picture below.



Click on the back-arrow in the top-left to go back to your skill.

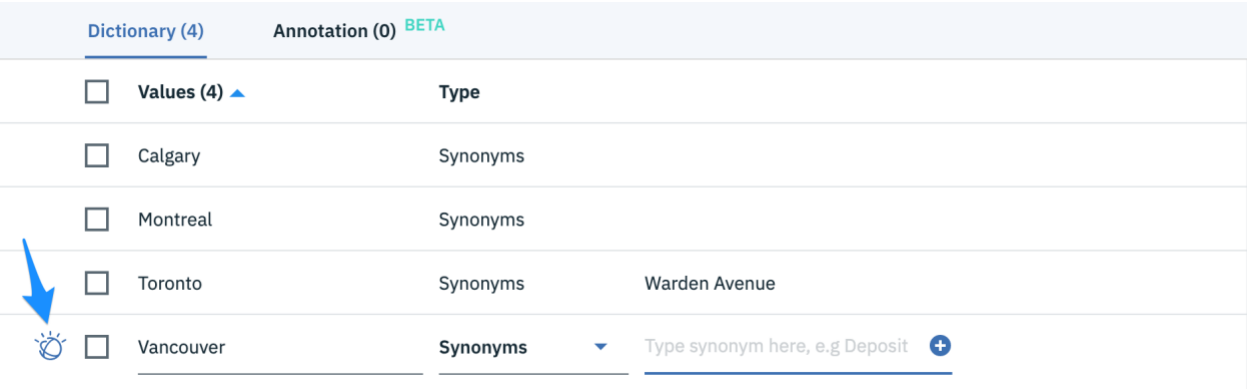
Open the *Try it out* panel and wait for Watson to finish training. What happens if you try, `hours of operation of your warden ave store` in the *Try it out* panel? Even though we haven't entered `Warden Avenue` spelled exactly as defined in the synonyms, fuzzy matching helps our chatbot detect the right entity value. It's worth noting that entity values can also have patterns, accessible from the *Synonyms* drop-down, as shown in the image below.



A pattern is an advanced feature that allows you to detect an entity value based not on a specific string (e.g., its synonym) but rather on a specific pattern like a properly formatted phone number, email address, or website address. If you are a programmer, it's worth noting that you specify your pattern as a Regular Expressions (e.g., `^\(?([0-9]{3})\)?[-. ]?([0-9]{3})[-. ]?([0-9]{4})$` to detect that a North American phone number was provided). If you are not a programmer, you can safely ignore this advanced feature. 😊

5. At any time you can click on an entity value to edit its name or synonym. Entities values are allowed to have spaces in them. When you first create an entity value, you're given the option to click on the *Show recommendations* button to select some synonyms from a list provided by Watson.

Try out this feature. Click on *Entities*, click on the `@location` entity, then select the @location:`Vancouver` entity value by clicking on it. A Watson icon will appear. Click on it as shown in the picture below.



Watson will make a few suggestions. For example, for Vancouver, it will suggest a few nearby cities as well as other major cities in Canada. **Select burnaby as a synonym** and then **click the *Add selected* button.** Finally, click on the X icon to close the recommendation section.

6. Use the *Try it out* panel to **test out these entity values**. Try entering, `What are your hours of operation in Montreal` and `Where is your Burnaby store located?` to see how Watson classifies that user question in terms of intents and entities.

7. Awesome. We can now recognize the intent as well as the cities corresponding to our stores. But what happens when the user enters `hours for Seattle` or `hours for Mumbai` where we don't have a store? **Try it and see**.

You'll notice that since we don't have an entity value for Seattle or Mumbai, neither will be picked up as an entity value. This is not necessarily a problem, because we can structure our chatbot to provide a generic, informative response if no recognized location is indicated.

But what if we do want to detect all locations to provide a more customized response? (e.g., "Unfortunately, we don't have a store in Seattle..."). We'd need an impractical large entity of our own that includes a list of all major cities.

As we'll see in Exercise 2, this is something that can instead be easily achieved with a system entity.

8. **Practice creating a new entity of your choice with some values, and then deleting it.** The process is very similar to that of intents.

Inside of *Entities,* you would select the checkmark next to the entity you created and no longer want, and then click the *Delete* button that appears.

Don't delete `@location` as we'll need it in our chatbot.

Exercise 2: Enable system entities

System entities allow us to easily detect common specific pieces of information like dates, times, numbers, currencies, etc. And among these, as anticipated in the previous exercise... locations. That's right. There is a `@sys-location` entity that will detect locations for us. One that would handle any city (or state, country, etc.).

So, in theory, we don't even need our `@location` entity, we could just use `@sys-location`. Well, not so fast. There are a couple of limitations here:

- You can't have synonyms for the cities detected by `@sys-location`. This is generally not a big deal and you could create your own entity for the specific cities that have synonyms and then check for both in the dialog (which we cover in Module 4).
- Fuzzy matching is not currently available for the location system entity in every dialog skill. In my experience, some Watson Assistant instances will have the ability to detect typos in the location system entities while others will not.

Lab 4: Create Entities

Generally speaking, it's worth using a system entity if one fits the bill for what you are trying to do. But if it makes your life more difficult due to your specific requirements, you're better off creating your own custom entity as we did in exercise 1.

Let's see this system entity in action.

1. To enable `@sys-location` click on *Entities* in the left menu, and **then on *System entities* underneath**. **Turn on @sys-location** by toggling it on under the *Status* column.

Take a moment to see what other system entities exist. If you open the *Try it out* panel quickly enough, you'll notice that Watson is now training on this large list of built-in locations.

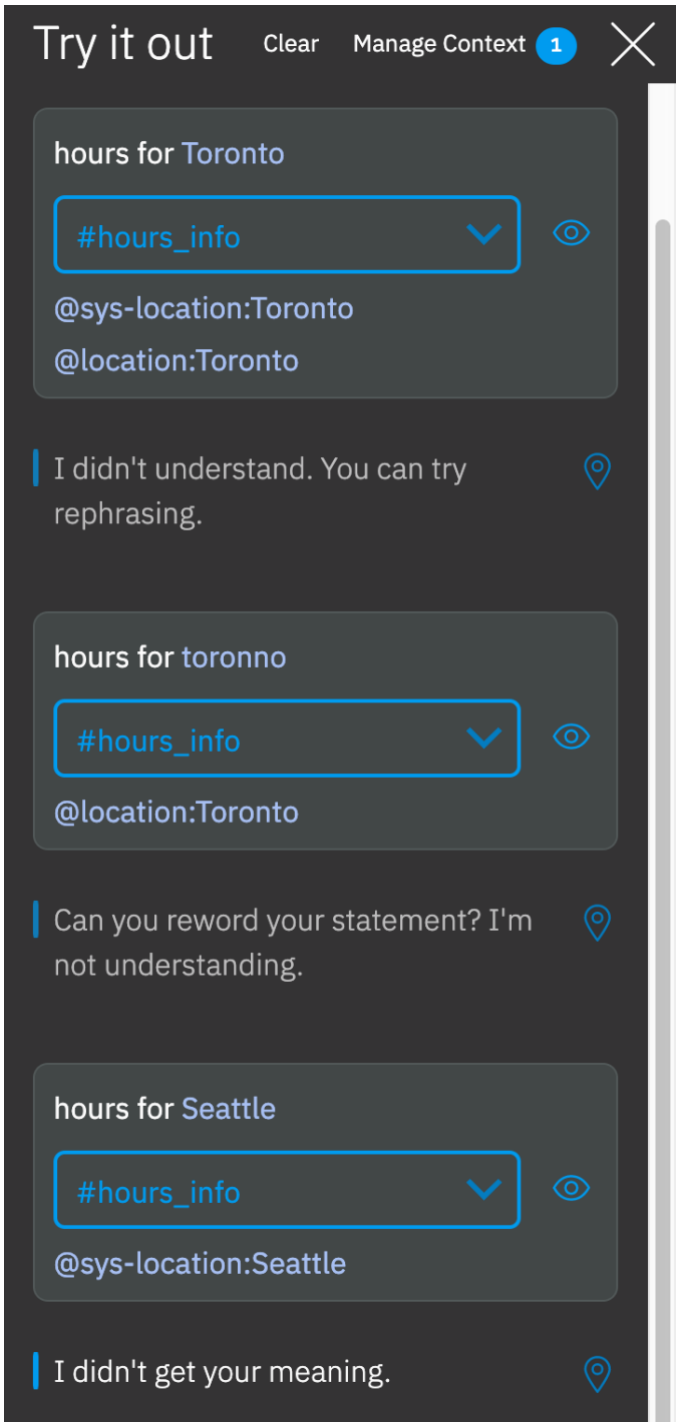
2. Wait for it to finish training and then **try entering `hours for Toronto`**. Which entity values are recognized?

3. You'll notice that both `@sys-location:Toronto` and `@location:Toronto` are detected. So far so good. **Now try, `hours for toronno`** (note the typo). Which entities are detected?

4. Some of you will continue to see both entities being detected. Others will see just `@location:Toronto` detected.

Depending on the version of the system entities deployed in your skill (behind the scenes and outside of your control), the system entity might expect an exact match for cities.

So, on one hand, we get to detect other cities and locations (**try `hours for Seattle`** again). On the other hand, the city has to be properly spelled by the user if your version of system entities fails to detect misspellings.



Depending on your chatbot, one or the other is a bigger compromise. In our case, we'll take the best of both worlds by keeping both. We'll then use `@location` for cities in which we have a shop, and `@sys-location` for other random cities the user might ask us about.

It's an unconventional approach (and arguably a bit overkill) but it will allow us to really fine tune our response to the user, as we'll see later on in the course.