In [ ]:
```
#### Virtual assistants and  accessing data
```

In [ ]:
```
## SQL statements in Python
```

In [4]:
```
# Import sqlite3
import sqlite3

# Open connection to DB
conn = sqlite3.connect('hotels.db')

# Create a cursor
c = conn.cursor()

# Define area and price
location, price = "north", "mid"
t = (location, price)

# Execute the query
c.execute('SELECT * FROM hotels WHERE location=? AND price=?', t)

# Print the results
print(c.fetchall())
```

```
[('Hotel California', 'mid', 'north', 3)]
```

In [ ]:
```
## Creating queries from parameters
```

In [12]:
```
# Define find_hotels()
def find_hotels(params):
    # Create the base query
    query = 'SELECT * FROM hotels'
    # Add filter clauses for each of the parameters
    if len(params) > 0:
        filters = ["{}=?".format(k) for k in params]
        query += " WHERE " + " and ".join(filters)
    # Create the tuple of values
    t = tuple(params.values())

    # Open connection to DB
    conn = sqlite3.connect("hotels.db")
    # Create a cursor
    c = conn.cursor()
    # Execute the query
    c.execute(query,t)
    # Return the results
    return c.fetchall()
```

In [19]:
```
## Using your custom function to find hotels
```

```
SELECT * FROM hotels
SELECT * FROM hotels WHERE location=?
[('Hotel California', 'mid', 'north', 3), ('Bens BnB', 'hi', 'north', 4)]
```

In [13]:
```
# Create the dictionary of column names and values
params = {"location":"north","price":"hi"}

# Find the hotels that match the parameters
print(find_hotels(params))
```

```
[('Bens BnB', 'hi', 'north', 4)]
```

In [ ]:
```
## Creating SQL from natural language
```

In [1]:

```python
# Import necessary modules
from rasa.nlu.training_data import load_data
from rasa.nlu.config import RasaNLUModelConfig
from rasa.nlu.model import Trainer
from rasa.nlu import config

# Create a trainer that uses this config
trainer = Trainer(config.load("config_spacy.yml"))

# Load the training data
training_data = load_data('demo-rasa.json')

# Create an interpreter by training the model
interpreter = trainer.train(training_data)

responses = [
    "I'm sorry :( I couldn't find anything like that",
    '{} is a great hotel!',
    '{} or {} would work!',
    '{} is one option, but I know others too :)'
]
```

```
WARNING:tensorflow:
The TensorFlow contrib module will not be included in TensorFlow 2.0.
For more information, please see:
  * https://github.com/tensorflow/community/blob/master/rfcs/20180907-contrib-sunset.md (https://github.com/tensorflow/community/blob/master/rfcs/20180907-contrib-sunset.md)
  * https://github.com/tensorflow/addons (https://github.com/tensorflow/addons)
  * https://github.com/tensorflow/io (https://github.com/tensorflow/io) (for I/O related ops)
If you depend on functionality not listed there, please file an issue.

WARNING:tensorflow:From C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\tensor2tensor\utils\adafactor.py:27: The name tf.train.Optimizer is deprecated. Please use tf.compat.v1.train.Optimizer instead.

WARNING:tensorflow:From C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\tensor2tensor\utils\multistep_optimizer.py:32: The name tf.train.AdamOptimizer is deprecated. Please use tf.compat.v1.train.AdamOptimizer instead.

WARNING:tensorflow:From C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\tensor2tensor\models\research\glow_init_hook.py:25: The name tf.train.SessionRunHook is deprecated. Please use tf.estimator.SessionRunHook instead.

WARNING:tensorflow:From C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\tensor2tensor\models\research\neural_stack.py:51: The name tf.nn.rnn_cell.RNNCell is deprecated. Please use tf.compat.v1.nn.rnn_cell.RNNCell instead.

WARNING:tensorflow:From C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\tensor2tensor\utils\trainer_lib.py:111: The name tf.OptimizerOptions is deprecated. Please use tf.compat.v1.OptimizerOptions instead.


WARNING:tensorflow:From C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\tensor2tensor\utils\trainer_lib.py:111: The name tf.OptimizerOptions is deprecated. Please use tf.compat.v1.OptimizerOptions instead.


WARNING:tensorflow:From C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\tensorflow_gan\python\estimator\tpu_gan_estimator.py:42: The name tf.estimator.tpu.TPUEstimator is deprecated. Please use tf.compat.v1.estimator.tpu.TPUEstimator instead.


WARNING:tensorflow:From C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\tensorflow_gan\python\estimator\tpu_gan_estimator.py:42: The name tf.estimator.tpu.TPUEstimator is deprecated. Please use tf.compat.v1.estimator.tpu.TPUEstimator instead.


WARNING:tensorflow:From C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\rasa\nlu\classifiers\embedding_intent_classifier.py:749: The name tf.set_random_seed is deprecated. Please use tf.compat.v1.set_random_seed instead.


C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\rasa\utils\common.py:351: UserWarning: Intent 'None' has only 1 training examples! Minimum is 2, training may fail.
C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\rasa\utils\common.py:351: UserWarning: Entity 'area' has only 1 training examples! The minimum is 2, because of this the training may fail.
WARNING:tensorflow:From C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\rasa\nlu\classifiers\embedding_intent_classifier.py:749: The name tf.set_random_seed is deprecated. Please use tf.compat.v1.set_random_seed instead.


WARNING:tensorflow:From C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\rasa\nlu\classifiers\embedding_intent_classifier.py:752: The name tf.placeholder is deprecated. Please use tf.compat.v1.placeholder instead.


WARNING:tensorflow:From C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\rasa\nlu\classifiers\embedding_intent_classifier.py:752: The name tf.placeholder is deprecated. Please use tf.compat.v1.placeholder instead.


WARNING:tensorflow:From C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\rasa\utils\train_utils.py:518: The name tf.data.Iterator is deprecated. Please use tf.compat.v1.data.Iterator instead.


WARNING:tensorflow:From C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\rasa\utils\train_utils.py:518: The name tf.data.Iterator is deprecated. Please use tf.compat.v1.data.Iterator instead.


WARNING:tensorflow:From C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\rasa\utils\train_utils.py:519: DatasetV1.output_types (from tensorflow.python.data.ops.dataset_ops) is deprecated and will be removed in a future version.
```

```
Instructions for updating:
Use `tf.compat.v1.data.get_output_types(dataset)`.

WARNING:tensorflow:From C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\rasa\utils\train_utils.py:519: DatasetV1.
output_types (from tensorflow.python.data.ops.dataset_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Use `tf.compat.v1.data.get_output_types(dataset)`.

WARNING:tensorflow:From C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\rasa\utils\train_utils.py:519: DatasetV1.
output_shapes (from tensorflow.python.data.ops.dataset_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Use `tf.compat.v1.data.get_output_shapes(dataset)`.

WARNING:tensorflow:From C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\rasa\utils\train_utils.py:519: DatasetV1.
output_shapes (from tensorflow.python.data.ops.dataset_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Use `tf.compat.v1.data.get_output_shapes(dataset)`.

WARNING:tensorflow:From C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\tensorflow_core\python\data\ops\iterator_
ops.py:347: Iterator.output_types (from tensorflow.python.data.ops.iterator_ops) is deprecated and will be removed in a fu
ture version.
Instructions for updating:
Use `tf.compat.v1.data.get_output_types(iterator)`.

WARNING:tensorflow:From C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\tensorflow_core\python\data\ops\iterator_
ops.py:347: Iterator.output_types (from tensorflow.python.data.ops.iterator_ops) is deprecated and will be removed in a fu
ture version.
Instructions for updating:
Use `tf.compat.v1.data.get_output_types(iterator)`.

WARNING:tensorflow:From C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\tensorflow_core\python\data\ops\iterator_
ops.py:348: Iterator.output_shapes (from tensorflow.python.data.ops.iterator_ops) is deprecated and will be removed in a f
uture version.
Instructions for updating:
Use `tf.compat.v1.data.get_output_shapes(iterator)`.

WARNING:tensorflow:From C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\tensorflow_core\python\data\ops\iterator_
ops.py:348: Iterator.output_shapes (from tensorflow.python.data.ops.iterator_ops) is deprecated and will be removed in a f
uture version.
Instructions for updating:
Use `tf.compat.v1.data.get_output_shapes(iterator)`.

WARNING:tensorflow:From C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\tensorflow_core\python\data\ops\iterator_
ops.py:350: Iterator.output_classes (from tensorflow.python.data.ops.iterator_ops) is deprecated and will be removed in a
future version.
Instructions for updating:
Use `tf.compat.v1.data.get_output_classes(iterator)`.

WARNING:tensorflow:From C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\tensorflow_core\python\data\ops\iterator_
ops.py:350: Iterator.output_classes (from tensorflow.python.data.ops.iterator_ops) is deprecated and will be removed in a
future version.
Instructions for updating:
Use `tf.compat.v1.data.get_output_classes(iterator)`.

WARNING:tensorflow:From C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\rasa\nlu\classifiers\embedding_intent_cla
ssifier.py:766: The name tf.placeholder_with_default is deprecated. Please use tf.compat.v1.placeholder_with_default inste
ad.


WARNING:tensorflow:From C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\rasa\nlu\classifiers\embedding_intent_cla
ssifier.py:766: The name tf.placeholder_with_default is deprecated. Please use tf.compat.v1.placeholder_with_default inste
ad.


WARNING:tensorflow:From C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\rasa\utils\train_utils.py:548: The name t
f.variable_scope is deprecated. Please use tf.compat.v1.variable_scope instead.


WARNING:tensorflow:From C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\rasa\utils\train_utils.py:548: The name t
f.variable_scope is deprecated. Please use tf.compat.v1.variable_scope instead.


WARNING:tensorflow:From C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\rasa\utils\train_utils.py:548: The name t
f.AUTO_REUSE is deprecated. Please use tf.compat.v1.AUTO_REUSE instead.


WARNING:tensorflow:From C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\rasa\utils\train_utils.py:548: The name t
f.AUTO_REUSE is deprecated. Please use tf.compat.v1.AUTO_REUSE instead.


WARNING:tensorflow:From C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\rasa\utils\train_utils.py:550: The name t
f.get_variable is deprecated. Please use tf.compat.v1.get_variable instead.


WARNING:tensorflow:From C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\rasa\utils\train_utils.py:550: The name t
f.get_variable is deprecated. Please use tf.compat.v1.get_variable instead.


WARNING:tensorflow:From C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\rasa\utils\train_utils.py:559: The name t
f.sparse.matmul is deprecated. Please use tf.sparse.sparse_dense_matmul instead.


WARNING:tensorflow:From C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\rasa\utils\train_utils.py:559: The name t
f.sparse.matmul is deprecated. Please use tf.sparse.sparse_dense_matmul instead.
```

```
WARNING:tensorflow:From C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\rasa\utils\train_utils.py:603: dense
(from tensorflow.python.layers.core) is deprecated and will be removed in a future version.
Instructions for updating:
Use keras.layers.Dense instead.

WARNING:tensorflow:From C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\rasa\utils\train_utils.py:603: dense (fro
m tensorflow.python.layers.core) is deprecated and will be removed in a future version.
Instructions for updating:
Use keras.layers.Dense instead.

WARNING:tensorflow:From C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\tensorflow_core\python\layers\core.py:18
7: Layer.apply (from tensorflow.python.keras.engine.base_layer) is deprecated and will be removed in a future version.
Instructions for updating:
Please use `layer.__call__` method instead.

WARNING:tensorflow:From C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\tensorflow_core\python\layers\core.py:18
7: Layer.apply (from tensorflow.python.keras.engine.base_layer) is deprecated and will be removed in a future version.
Instructions for updating:
Please use `layer.__call__` method instead.

WARNING:tensorflow:From C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\rasa\utils\train_utils.py:605: dropout (f
rom tensorflow.python.layers.core) is deprecated and will be removed in a future version.
Instructions for updating:
Use keras.layers.dropout instead.

WARNING:tensorflow:From C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\rasa\utils\train_utils.py:605: dropout (f
rom tensorflow.python.layers.core) is deprecated and will be removed in a future version.
Instructions for updating:
Use keras.layers.dropout instead.

WARNING:tensorflow:From C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\tensorflow_core\python\util\dispatch.py:1
80: batch_gather (from tensorflow.python.ops.array_ops) is deprecated and will be removed after 2017-10-25.
Instructions for updating:
`tf.batch_gather` is deprecated, please use `tf.gather` with `batch_dims=-1` instead.

WARNING:tensorflow:From C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\tensorflow_core\python\util\dispatch.py:1
80: batch_gather (from tensorflow.python.ops.array_ops) is deprecated and will be removed after 2017-10-25.
Instructions for updating:
`tf.batch_gather` is deprecated, please use `tf.gather` with `batch_dims=-1` instead.

WARNING:tensorflow:From C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\rasa\utils\train_utils.py:984: The name t
f.losses.softmax_cross_entropy is deprecated. Please use tf.compat.v1.losses.softmax_cross_entropy instead.


WARNING:tensorflow:From C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\rasa\utils\train_utils.py:984: The name t
f.losses.softmax_cross_entropy is deprecated. Please use tf.compat.v1.losses.softmax_cross_entropy instead.


WARNING:tensorflow:From C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\tensorflow_core\python\ops\losses\losses_
impl.py:121: where (from tensorflow.python.ops.array_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where

WARNING:tensorflow:From C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\tensorflow_core\python\ops\losses\losses_
impl.py:121: where (from tensorflow.python.ops.array_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where

WARNING:tensorflow:From C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\rasa\utils\train_utils.py:986: The name t
f.losses.get_regularization_loss is deprecated. Please use tf.compat.v1.losses.get_regularization_loss instead.


WARNING:tensorflow:From C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\rasa\utils\train_utils.py:986: The name t
f.losses.get_regularization_loss is deprecated. Please use tf.compat.v1.losses.get_regularization_loss instead.


WARNING:tensorflow:From C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\rasa\nlu\classifiers\embedding_intent_cla
ssifier.py:774: The name tf.Session is deprecated. Please use tf.compat.v1.Session instead.


WARNING:tensorflow:From C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\rasa\nlu\classifiers\embedding_intent_cla
ssifier.py:774: The name tf.Session is deprecated. Please use tf.compat.v1.Session instead.


WARNING:tensorflow:From C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\rasa\utils\train_utils.py:1168: The name
tf.global_variables_initializer is deprecated. Please use tf.compat.v1.global_variables_initializer instead.


WARNING:tensorflow:From C:\Users\84353\Anaconda3\envs\chat_box\lib\site-packages\rasa\utils\train_utils.py:1168: The name
tf.global_variables_initializer is deprecated. Please use tf.compat.v1.global_variables_initializer instead.

Epochs: 100%|██████████████████████████████████████████████████████████████████████████████| 300/300
[00:07<00:00, 40.59it/s, loss=0.926, acc=1.000]
```

In [14]:
```
# Define respond()
def respond(message):
    # Extract the entities
    entities = interpreter.parse(message)["entities"]
    # Initialize an empty params dictionary
    params = {}
    # Fill the dictionary with entities
    for ent in entities:
        params[ent["entity"]] = str(ent["value"])

    # Find hotels that match the dictionary
    results = find_hotels(params)
    # Get the names of the hotels and index of the response
    names = [r[0] for r in results]
    n = min(len(results),3)
    # Select the nth element of the responses array
    return responses[n].format(*names)

print(respond("I want an expensive hotel in the south town"))
```

```
Grand Hotel is a great hotel!
```

In [ ]:
```
#### Incremental slot filling and negation
```

In [ ]:
```
## Refining your search
```

In [15]:
```
# Define a respond function, taking the message and existing params as input
def respond(message, params):
    # Extract the entities
    entities = interpreter.parse(message)["entities"]
    # Fill the dictionary with entities
    for ent in entities:
        params[ent["entity"]] = str(ent["value"])

    # Find the hotels
    results = find_hotels(params)
    names = [r[0] for r in results]
    n = min(len(results), 3)
    # Return the appropriate response
    return responses[n].format(*names), params

# Initialize params dictionary
params = {}

# Pass the messages to the bot
for message in ["I want an expensive hotel", "in the north of town"]:
    print("USER: {}".format(message))
    response, params = respond(message, params)
    print("BOT: {}".format(response))
```

```
USER: I want an expensive hotel
BOT: Grand Hotel is one option, but I know others too :)
USER: in the north of town
BOT: Bens BnB is a great hotel!
```

In [ ]:
```
## Basic negation
```

In [16]:
```
tests= [
    ("no I don't want to be in the south", {'south': False}),
    ('no it should be in the south', {'south': True}),
    ('no in the south not the north', {'north': False, 'south': True}),
    ('not north', {'north': False})
]
```

In [20]:
```python
# Define negated_ents()
def negated_ents(phrase):
    # Extract the entities using keyword matching
    ents = [e for e in ["north", "south"] if e in phrase]
    # Find the index of the final character of each entity
    ends = sorted([phrase.find(e) + len(e) for e in ents])
    # Initialise a list to store sentence chunks
    chunks = []
    # Take slices of the sentence up to and including each entitiy
    start = 0
    for end in ends:
        chunks.append(phrase[start:end])
        start = end
    result = {}
    # Iterate over the chunks and look for entities
    for chunk in chunks:
        for ent in ents:
            if ent in chunk:
                # If the entity is preceeded by a negation, give it the key False
                if "not" in chunk or "n't" in chunk:
                    result[ent] = False
                else:
                    result[ent] = True
    return result

# Check that the entities are correctly assigned as True or False
for test in tests:
    print(negated_ents(test[0]) == test[1])
```

```
True
True
True
True
```

In [ ]:
```python
## Filtering with excluded slots
```

In [21]:
```python
def negated_ents(phrase, ent_vals):
    ents = [e for e in ent_vals if e in phrase]
    ends = sorted([phrase.index(e) + len(e) for e in ents])
    start = 0
    chunks = []
    for end in ends:
        chunks.append(phrase[start:end])
        start = end
    result = {}
    for chunk in chunks:
        for ent in ents:
            if ent in chunk:
                if "not" in chunk or "n't" in chunk:
                    result[ent] = False
                else:
                    result[ent] = True
    return result

def find_hotels(params, neg_params):
    query = 'SELECT * FROM hotels'
    if len(params) > 0 and len(neg_params) > 0:
        filters = ["{}=?".format(k) for k in params] + ["{}!=?".format(k) for k in neg_params]
        query += " WHERE " + " and ".join(filters)
    elif len(neg_params) > 0:
        filters = ["{}!=?".format(k) for k in neg_params]
        query += " WHERE " + " and ".join(filters)
    elif len(params) > 0:
        filters = ["{}=?".format(k) for k in params]
        query += " WHERE " + " and ".join(filters)

    t = tuple(dict(list(params.items()) + list(neg_params.items())).values())
    # open connection to DB
    conn = sqlite3.connect('hotels.db')
    # create a cursor
    c = conn.cursor()
    c.execute(query, t)
    return c.fetchall()
```

In [22]:

```
# Define the respond function
def respond(message,params,neg_params):
    # Extract the entities
    entities = interpreter.parse(message)["entities"]
    ent_vals = [e["value"] for e in entities]
    # Look for negated entities
    negated = negated_ents(message,ent_vals)
    for ent in entities:
        if ent["value"] in negated and not negated[ent["value"]]:
            neg_params[ent["entity"]] = str(ent["value"])
        else:
            params[ent["entity"]] = str(ent["value"])
    # Find the hotels
    results = find_hotels(params,neg_params)
    names = [r[0] for r in results]
    n = min(len(results),3)
    # Return the correct response
    return responses[n].format(*names), params, neg_params

# Initialize params and neg_params
params = {}
neg_params = {}

# Pass the messages to the bot
for message in ["I want an expensive hotel", "but not in the south of town"]:
    print("USER: {}".format(message))
    response, params, neg_params = respond(message, params, neg_params)
    print("BOT: {}".format(response))

```

```
USER: I want an expensive hotel
BOT: Grand Hotel is one option, but I know others too :)
USER: but not in the south of town
BOT: Bens BnB or The Grand would work!
```