In [52]:

```
###################################
# Bespoke Chatbot NLP Workshop 2019-07-22
###################################

from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import precision_recall_fscore_support

# Train
training_phrases = {
    'when_is_check_in' : ' '.join([
        'when is check-in',
        'When can I check in?',
        'whens checkin'
    ]),
    'where_is_the_front_desk' : ' '.join([
        'Where is the front desk?',
        'what is the location of the front desk?'
    ])
}

training_documents = list(training_phrases.values())
labels = list(training_phrases.keys())

vectorizer = CountVectorizer()
X = vectorizer.fit_transform(training_documents)
classifier = MultinomialNB()
classifier.fit(X, labels)

# Predict
def predict(raw_queries):
    queries = vectorizer.transform(raw_queries)
    return classifier.predict(queries) # predict_proba

raw_queries = ["where location", "when is", "where is check in location"]
predicted = predict(raw_queries)

# Evaluate
expected = ["where_is_the_front_desk", "when_is_check_in", "where_is_the_front_desk"

evaluation = precision_recall_fscore_support(expected, predicted, average='micro')
metrics = {}
(metrics['p'], metrics['r'], metrics['f1'], _) = evaluation
print("Evaluation metrics: ", metrics)

# Retrieve answer
answers = {
    'when_is_check_in' : 'Check in is at 3pm! :)',
    'where_is_the_front_desk' : 'The front desk is located on the 2nd floor.',
}

user_query = 'where is my hotel please?'
predicted = predict([user_query])
print("Answer to user query: ", answers[predicted[0]])
```

```
Evaluation metrics:  {'p': 0.6666666666666666, 'r': 0.666666666666666
6, 'f1': 0.6666666666666666}
Answer to user query:  The front desk is located on the 2nd floor.
```