In [1]:
```
import os
import csv
import pandas as pd
import numpy as np
import sklearn
import string
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.decomposition import NMF
from nltk import tokenize
```

In [2]:
```
def print_top_words(model, feature_names, n_top_words):
    for topic_idx, topic in enumerate(model.components_):
        message = "Topic #%d: " % topic_idx
        message += " ".join([feature_names[i]
                             for i in topic.argsort()[:-n_top_words - 1:-1]])
        print(message)
    print()
```

In [3]:
```
def display_topics(model, feature_names, num_topics, no_top_words):
    for topic_idx, topic in enumerate(model.components_):
        if topic_idx < num_topics:
            print("{:11}".format("Topic %d:" %(topic_idx)), end='')
            print(", ".join(['{:04.3f}*'.format(topic[i])+feature_names[i] \
                             for i in topic.argsort()[:-no_top_words-1:-1]]))
```

In [4]:
```
# Read in Data
data = pd.read_csv('hash_house.csv')
data['userid'] = data['Unnamed: 0']
data.head()
```

Out[4]:

	Unnamed: 0	name	stars_y	text	userid
0	0	Hash House A Go Go	5	Firstly, this restaurant is in The Linq Hotel,...	0
1	1	Hash House A Go Go	4	This place had monsterous proportions OMG! One...	1
2	2	Hash House A Go Go	5	This place freaking rocks. Must go to when in ...	2
3	3	Hash House A Go Go	3	Visited HHAGG ago go for the first time on 5/5...	3
4	4	Hash House A Go Go	3	Big portions. Sharing is highly recommended. H...	4

In [5]:
```
# Split reviews into individual sentences
df = pd.DataFrame(columns=['userid','sentence','stars'])
for i in range(0,len(data),1):
    sentences = tokenize.sent_tokenize(data.text[i])
    for j in sentences:
        df = df.append({'userid':data.userid[i],'sentence':j,'stars':data.stars_y[i]},ignore_index=Tru
e)
```

In [6]:
```
df.head()
```

Out[6]:

	userid	sentence	stars
0	0	Firstly, this restaurant is in The Linq Hotel,...	5
1	0	Expect a line.	5
2	0	Waited only about 15 minutes to be seated, tho...	5
3	0	Greeted by Tony our waiter who was really warm...	5
4	0	Ordered the Sage Fried Chicken and Waffles.	5

In [7]:
```
# Create Corpus for TFIDF
corpus = []
for i in df.sentence:
        corpus.append(i)
```

7 Topics

In [8]:
```
n_components = 7
n_top_words = 15

# TFIDF Vectorizer
tfidf_vectorizer = TfidfVectorizer(stop_words='english')
tfidf = tfidf_vectorizer.fit_transform(corpus)

# NMF reduction
nmf = NMF(n_components=n_components).fit(tfidf)
W_pos = nmf.fit_transform(tfidf)

# Output Topics
print("\nTopics in NMF model (generalized Kullback-Leibler divergence):")
tfidf_feature_names = tfidf_vectorizer.get_feature_names()
print_top_words(nmf, tfidf_feature_names, n_top_words)
```

```
Topics in NMF model (generalized Kullback-Leibler divergence):
Topic #0: great service friendly excellent experience staff customer slow server fast atmosphere atte
ntive waiter quick bad
Topic #1: chicken waffles fried sage benedict ordered bacon got eggs delicious andy waffle potatoes c
rispy hash
Topic #2: huge portions large big share portion delicious people prices plate massive enormous hungry
meal tasty
Topic #3: good really pretty service overall just potatoes biscuits bloody thing mary taste coffee bi
scuit wasn
Topic #4: place vegas breakfast definitely hash love house try time come eat best recommend just las
Topic #5: food amazing delicious man vs awesome just came lot price excellent took quality tasty larg
e
Topic #6: wait worth long time minutes hour seated 30 table minute 45 20 come definitely 10
```

- Topic #0: Service
- Topic #1: Food
- Topic #2: "Worth it"
- Topic #3: Food / Service
- Topic #4:
- Topic #5: Food
- Topic #6: Wait

In [70]:
```
# Append Topic with highest score
array = []
# For all NMF array
for i in W_pos:
    # Create dictionary with Topics and its NMF scores for each sentence
    topic_dict = {}
    for ind, w in enumerate(i):
        topic_dict[ind] = w
    # Classify sentence to the topic with highest score
    array.append(max(topic_dict, key=topic_dict.get))
# Create new column in df for topic
df['Topic'] = array
```

In [71]:
```
df.head()
```

Out[71]:

	userid	sentence	stars	Topic
0	0	Firstly, this restaurant is in The Linq Hotel,...	5	4
1	0	Expect a line.	5	6
2	0	Waited only about 15 minutes to be seated, tho...	5	6
3	0	Greeted by Tony our waiter who was really warm...	5	3
4	0	Ordered the Sage Fried Chicken and Waffles.	5	1