

## My Project

Generated by Doxygen 1.8.11



# Contents

<b>1</b>	<b>Voice Enabled Natural Language Search Engine for Yelp</b>	<b>1</b>
<b>2</b>	<b>Namespace Index</b>	<b>3</b>
2.1	Namespace List . . . . .	3
<b>3</b>	<b>Hierarchical Index</b>	<b>5</b>
3.1	Class Hierarchy . . . . .	5
<b>4</b>	<b>Class Index</b>	<b>7</b>
4.1	Class List . . . . .	7
<b>5</b>	<b>File Index</b>	<b>9</b>
5.1	File List . . . . .	9
<b>6</b>	<b>Namespace Documentation</b>	<b>11</b>
6.1	distance Namespace Reference . . . . .	11
6.1.1	Detailed Description . . . . .	11
6.1.2	Function Documentation . . . . .	11
6.1.2.1	distance(p1, p2) . . . . .	11
6.1.2.2	toRad(degree) . . . . .	11
6.2	flask_jqry Namespace Reference . . . . .	11
6.2.1	Function Documentation . . . . .	12
6.2.1.1	main() . . . . .	12
6.2.1.2	process(q) . . . . .	12
6.2.1.3	view_do_something() . . . . .	12
6.2.2	Variable Documentation . . . . .	12

6.2.2.1	app	12
6.2.2.2	searcher	12
6.3	indexer Namespace Reference	12
6.4	query Namespace Reference	12
6.4.1	Detailed Description	13
6.4.2	Function Documentation	13
6.4.2.1	all_cities()	13
6.4.2.2	extract_entity_names(t)	13
6.4.2.3	parse_filters(original_setence)	13
6.4.2.4	parse_location(original_setence)	14
6.4.2.5	parse_time(original_setence)	14
6.4.2.6	query(original)	14
6.4.3	Variable Documentation	14
6.4.3.1	Cities	14
6.4.3.2	latitude	14
6.4.3.3	longitude	14
6.4.3.4	nums	14
6.4.3.5	States	14
6.4.3.6	States_abbr	14
6.4.3.7	times	14
6.5	query_speechrecognition Namespace Reference	14
6.5.1	Detailed Description	15
6.5.2	Function Documentation	15
6.5.2.1	all_cities()	15
6.5.2.2	extract_entity_names(t)	15
6.5.2.3	get_audio_query()	15
6.5.2.4	parse_filters(original_setence)	16
6.5.2.5	parse_location(original_setence)	16
6.5.2.6	parse_time(original_setence)	16
6.5.2.7	query_speech()	16

6.5.3	Variable Documentation	16
6.5.3.1	Cities	16
6.5.3.2	latitude	16
6.5.3.3	longitude	16
6.5.3.4	nums	16
6.5.3.5	States	16
6.5.3.6	States_abbrev	16
6.5.3.7	times	16
6.6	searcher Namespace Reference	16
6.6.1	Detailed Description	17
6.7	test_indexer Namespace Reference	17
6.7.1	Detailed Description	17
6.7.2	Function Documentation	17
6.7.2.1	main()	17
6.8	test_speech Namespace Reference	17
6.8.1	Detailed Description	17
6.8.2	Function Documentation	17
6.8.2.1	main()	17
<b>7</b>	<b>Class Documentation</b>	<b>19</b>
7.1	indexer.Indexer Class Reference	19
7.1.1	Detailed Description	19
7.1.2	Constructor & Destructor Documentation	20
7.1.2.1	__init__(self, source_path, destination=None)	20
7.1.3	Member Function Documentation	20
7.1.3.1	index(self)	20
7.1.3.2	preprocess(self)	20
7.1.4	Member Data Documentation	20
7.1.4.1	business	20
7.1.4.2	data	20
7.1.4.3	destination	20

7.1.4.4	review	20
7.1.4.5	source_path	20
7.1.4.6	tip	20
7.2	searcher.Searcher Class Reference	21
7.2.1	Constructor & Destructor Documentation	21
7.2.1.1	__init__(self, path)	21
7.2.2	Member Function Documentation	21
7.2.2.1	check_distance(self, id, value)	21
7.2.2.2	check_hours(self, id, time)	21
7.2.2.3	check_parking(self, id)	21
7.2.2.4	filtering(self, search_results, filters)	22
7.2.2.5	hour_to_number(self, hours)	22
7.2.2.6	process_query(self, query, filters)	22
7.2.2.7	searching(self, query)	22
7.2.3	Member Data Documentation	22
7.2.3.1	analyzer	22
7.2.3.2	data	22
7.2.3.3	reader	22
7.2.3.4	searcher	22
<b>8</b>	<b>File Documentation</b>	<b>23</b>
8.1	/Users/Federer-HYJ/Documents/EECS549/final project/YelpNLSearch/distance.py File Reference	23
8.2	/Users/Federer-HYJ/Documents/EECS549/final project/YelpNLSearch/flask_jqry.py File Reference	23
8.3	/Users/Federer-HYJ/Documents/EECS549/final project/YelpNLSearch/indexer.py File Reference	24
8.4	/Users/Federer-HYJ/Documents/EECS549/final project/YelpNLSearch/query.py File Reference	24
8.5	/Users/Federer-HYJ/Documents/EECS549/final project/YelpNLSearch/query_speechrecognition.py File Reference	24
8.6	/Users/Federer-HYJ/Documents/EECS549/final project/YelpNLSearch/README.md File Reference	25
8.7	/Users/Federer-HYJ/Documents/EECS549/final project/YelpNLSearch/searcher.py File Reference	25
8.8	/Users/Federer-HYJ/Documents/EECS549/final project/YelpNLSearch/static/js/bootstrap.min.js File Reference	25
8.8.1	Function Documentation	26

8.8.1.1	<a href="#">a(function(){a.support.transition=b()}})(jQuery)</a>	26
8.8.1.2	<a href="#">b()</a>	26
8.8.1.3	<a href="#">close(jQuery)</a>	26
8.8.1.4	<a href="#">hasClass("btn")"   "   (c</a>	26
8.8.1.5	<a href="#">if("undefined"==typeof jQuery) throw new Error("Bootstrap requires jQuery")</a>	26
8.8.2	<a href="#">Variable Documentation</a>	27
8.8.2.1	<a href="#">a</a>	27
8.8.2.2	<a href="#">alert</a>	27
8.8.2.3	<a href="#">b</a>	27
8.8.2.4	<a href="#">button</a>	27
8.8.2.5	<a href="#">c</a>	27
8.8.2.6	<a href="#">close</a>	27
8.8.2.7	<a href="#">Constructor</a>	27
8.8.2.8	<a href="#">d</a>	27
8.8.2.9	<a href="#">DEFAULTS</a>	27
8.8.2.10	<a href="#">emulateTransitionEnd</a>	27
8.8.2.11	<a href="#">noConflict</a>	27
8.8.2.12	<a href="#">setState</a>	27
8.8.2.13	<a href="#">toggle</a>	27
8.9	<a href="#">/Users/Federer-HYJ/Documents/EECS549/final project/YelpNLSearch/test_indexer.py File Reference</a>	27
8.10	<a href="#">/Users/Federer-HYJ/Documents/EECS549/final project/YelpNLSearch/test_speech.py File Reference</a>	28





## Chapter 1

# Voice Enabled Natural Language Search Engine for Yelp

### Dataset

This system runs on Yelp Dataset Challenge dataset ([https://www.yelp.com/dataset\\_challenge](https://www.yelp.com/dataset_challenge)). Download the dataset and extract the files to a folder.

### Packages/tools needed to run the system:

The following packages needs to be install



## Chapter 2

# Namespace Index

### 2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">distance</a>	11
<a href="#">flask_jqry</a>	11
<a href="#">indexer</a>	12
<a href="#">query</a>	12
<a href="#">query_speechrecognition</a>	14
<a href="#">searcher</a>	16
<a href="#">test_indexer</a>	17
<a href="#">test_speech</a>	17



## Chapter 3

# Hierarchical Index

### 3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

object	
indexer.Indexer . . . . .	<a href="#">19</a>
searcher.Searcher . . . . .	<a href="#">21</a>



## Chapter 4

# Class Index

### 4.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">indexer.Indexer</a>	19
<a href="#">searcher.Searcher</a>	21





## Chapter 5

# File Index

### 5.1 File List

Here is a list of all files with brief descriptions:

/Users/Federer-HYJ/Documents/EECS549/final project/YelpNLSearch/ <a href="#">distance.py</a> . . . . .	23
/Users/Federer-HYJ/Documents/EECS549/final project/YelpNLSearch/ <a href="#">flask_jqry.py</a> . . . . .	23
/Users/Federer-HYJ/Documents/EECS549/final project/YelpNLSearch/ <a href="#">indexer.py</a> . . . . .	24
/Users/Federer-HYJ/Documents/EECS549/final project/YelpNLSearch/ <a href="#">query.py</a> . . . . .	24
/Users/Federer-HYJ/Documents/EECS549/final project/YelpNLSearch/ <a href="#">query_speechrecognition.py</a> . . .	24
/Users/Federer-HYJ/Documents/EECS549/final project/YelpNLSearch/ <a href="#">searcher.py</a> . . . . .	25
/Users/Federer-HYJ/Documents/EECS549/final project/YelpNLSearch/ <a href="#">test_indexer.py</a> . . . . .	27
/Users/Federer-HYJ/Documents/EECS549/final project/YelpNLSearch/ <a href="#">test_speech.py</a> . . . . .	28
/Users/Federer-HYJ/Documents/EECS549/final project/YelpNLSearch/static/js/ <a href="#">bootstrap.min.js</a> . . . . .	25



## Chapter 6

# Namespace Documentation

### 6.1 distance Namespace Reference

#### Functions

- def `toRad` (degree)
- def `distance` (p1, p2)

#### 6.1.1 Detailed Description

```
calculate the distance between two lat lng points in miles
point: (lat, lng)

p1 = (42.280826, -83.743038) # ann arbor
p2 = (41.878114, -87.629798) # chicago
print distance(p1, p2)
```

#### 6.1.2 Function Documentation

##### 6.1.2.1 `def distance.distance ( p1, p2 )`

##### 6.1.2.2 `def distance.toRad ( degree )`

```
convert degree to radian
```

### 6.2 flask\_jqry Namespace Reference

#### Functions

- def `main` ()
- def `view_do_something` ()
- def `process` (q)

## Variables

- `app` = `Flask(__name__)`
- `searcher` = `None`

### 6.2.1 Function Documentation

#### 6.2.1.1 `def flask_jqry.main ( )`

This function is used for display the webpage on the browser

#### 6.2.1.2 `def flask_jqry.process ( q )`

Generating parsed queries and doing search

#### 6.2.1.3 `def flask_jqry.view_do_something ( )`

This function for handling the POST request sent from web page and call another python script to run for q

### 6.2.2 Variable Documentation

#### 6.2.2.1 `flask_jqry.app = Flask(__name__)`

#### 6.2.2.2 `flask_jqry.searcher = None`

## 6.3 indexer Namespace Reference

### Classes

- class `Indexer`

## 6.4 query Namespace Reference

### Functions

- `def all_cities ()`
- `def extract_entity_names (t)`
- `def parse_location (original_setence)`
- `def parse_time (original_setence)`
- `def parse_filters (original_setence)`
- `def query (original)`

## Variables

- `times` = `time.localtime()`
- list `nums`
- dictionary `Cities` = {}
- dictionary `States` = {}
- dictionary `States_abbr` = {}
- float `latitude` = 36.169941
- float `longitude` = -115.139830

### 6.4.1 Detailed Description

@This file is used to parse the query into the multi filters and modify the query sentence, so that the search engine is able to retrieve a list of business from dataset which meeting all requirements of query.

The filters include location limit, open time limit, happy hour, suitable for group, suitable for kids, outdoor table availability, various price level, take-out and delivery availability, reservation availability, and parking availability

The words shows in the filter will be removed from query string so that search engine is able to get a more fair and relevant ranking of different business.

@ Input: the string of query sentence

@ Output: the imporved query sentence and a list of filters.

@ Package used: nltk, time, geonamescache

### 6.4.2 Function Documentation

#### 6.4.2.1 `def query.all_cities ( )`

This function will set three dictionary of python.

The package geonamescache contains a list of cities in the World and a list of states in Unite States  
After scan the list of cities and the list of states, the name of cities and the name of states of America with all lower-case will be loaded into dictionary `Cities` and `States` separately.

#### 6.4.2.2 `def query.extract_entity_names ( t )`

This function is a helper function, which will be recursively called to separate one sentence into various words group. And from the groups of word, the city name and state name can be extracted as filter

#### 6.4.2.3 `def query.parse_filters ( original_sentence )`

This funciton will extract specified attribute of business listed in dataset, like happy hour, suitable for group, suitable for kids, outdoor table availability, various price level, take-out and delivery availability, reservation availability, and parking availability. The filters translated from the query will locat the specified business meeting all requirement

#### 6.4.2.4 `def query.parse_location ( original_setence )`

This function will extract location information from query sentence, including identified city name, state name, and the distance limit from the current location. If the query try to find near business, the default distance limit is 5 miles.

#### 6.4.2.5 `def query.parse_time ( original_setence )`

This function will extract time information from query sentence, including identified weekdays and time period like morning and afternoon. The query can also specify exact time like 5 o'clock and 3 pm.

#### 6.4.2.6 `def query.query ( original )`

This is main function for parsing query. The input is the query sentence and the outputs are modified query string and all filters

### 6.4.3 Variable Documentation

#### 6.4.3.1 dictionary `query.Cities = {}`

#### 6.4.3.2 float `query.laititude = 36.169941`

#### 6.4.3.3 float `query.longitude = -115.139830`

#### 6.4.3.4 list `query.num`s

Initial value:

```
1 = ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', \
2   'nigh', 'ten', 'eleven', 'twelve', 'thirtenn', 'fourteen', 'fifteen', \
3   'sixteen', 'seventeen', 'eighteen', 'nineteen', 'twenty']
```

#### 6.4.3.5 dictionary `query.States = {}`

#### 6.4.3.6 dictionary `query.States_abbr = {}`

#### 6.4.3.7 `query.times = time.localtime()`

## 6.5 `query_speechrecognition` Namespace Reference

### Functions

- `def all_cities ()`
- `def get_audio_query ()`
- `def extract_entity_names (t)`
- `def parse_location (original_setence)`
- `def parse_time (original_setence)`
- `def parse_filters (original_setence)`
- `def query_speech ()`

## Variables

- `times` = `time.localtime()`
- list `nums` = ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nigh', 'ten', 'eleven', 'twelve', 'thirtenn', 'fourteen', 'fifteen', 'sixteen', 'seventeen', 'eighteen', 'ninteen', 'twenty']
- dictionary `Cities` = {}
- dictionary `States` = {}
- dictionary `States_abbr` = {}
- float `latitude` = 36.169941
- float `longitude` = -115.139830

### 6.5.1 Detailed Description

This file is very similar to `query.py`. The only difference is that text input of the query sentence becomes audio input  
This file is used to parse the query into the multi filters and modify the query sentence, so that the search engine is able to retrieve a list of business from dataset which meeting all requirements of query.

The filters include location limit, open time limit, happy hour, suitable for group, suitable for kids, outdoor table availability, various price level, take-out and delivery availability, reservation availability, and parking availability

The words shows in the filter will be removed from query string so that search engine is able to get a more fair and relevant ranking of different business.

@ Input: the input would be audio

@ Output: the imporved query sentence and a list of filters.

@ Package used: `nltk`, `time`, `geonamescache`, `speech_recognition`

### 6.5.2 Function Documentation

#### 6.5.2.1 `def query_speechrecognition.all_cities ( )`

This function will set three dictionary of python.

The package `geonamescache` contians a list of cities in the World and a list of states in Unite States  
After scan the list of cities and the list of states, the name of cities and the name of states of America with all lower-case will be loaded into dictionary `Cities` and `States` separately.

#### 6.5.2.2 `def query_speechrecognition.extract_entity_names ( t )`

This function is a helper function, which will be recursively called to separate one sentence into various words group. And from the groups of word, the city name and state name can be extracted as filter

#### 6.5.2.3 `def query_speechrecognition.get_audio_query ( )`

This function wiill open the microphone and record the speech. Then the speech will be translated into text by google speedch recognition.

#### 6.5.2.4 `def query_speechrecognition.parse_filters ( original_setence )`

This funciton will extract specified attribute of business listed in dataset, like happy hour, suitable for group, suitable for kids, outdoor table availability, various price level, take-out and delivery availability, reservation availability, and parking availability. The filters translated from the query will locat the specified business meeting all requirement

#### 6.5.2.5 `def query_speechrecognition.parse_location ( original_setence )`

This funciton will extract location information from query sentence, including identified city name, state name, and the distance limit from the current location. If the query try to find near business, the default distance limit is 5 miles.

#### 6.5.2.6 `def query_speechrecognition.parse_time ( original_setence )`

This funciton will extract time information from query sentence, including identified weekdays and time period like morning and afternoon. The query can also specify exact time like 5 o'clock and 3 pm.

#### 6.5.2.7 `def query_speechrecognition.query_speech ( )`

This is main function for parsing query. The input is the query sentence and the outputs are modified query string and all filters

### 6.5.3 Variable Documentation

6.5.3.1 `dictionary query_speechrecognition.Cities = {}`

6.5.3.2 `float query_speechrecognition.laititude = 36.169941`

6.5.3.3 `float query_speechrecognition.longitude = -115.139830`

6.5.3.4 `list query_speechrecognition.nums = ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nigh', 'ten', 'eleven', 'twelve', 'thirtenn', 'fourteen', 'fifteen', 'sixteen', 'seventeen', 'eighteen', 'ninteen', 'twenty']`

6.5.3.5 `dictionary query_speechrecognition.States = {}`

6.5.3.6 `dictionary query_speechrecognition.States_abbr = {}`

6.5.3.7 `query_speechrecognition.times = time.localtime()`

## 6.6 searcher Namespace Reference

### Classes

- class [Searcher](#)



### 6.6.1 Detailed Description

This class is used to perform query searches.  
When a Searcher object is created, it loads the preprocessed data from disk to memory.  
A query can be performed by calling function `perform_query`. The function will return list of search results ranked by ranking score.

## 6.7 test\_indexer Namespace Reference

### Functions

- def `main` ()

### 6.7.1 Detailed Description

This script is used to index the yelp dataset.  
Please use the following command to start this script:  
`python test_indexer.py <path_to_yelp_dataset_folder> data.json`  
The inverted index will be saved to `./index/` folder, refer to `indexer.py` for more information.  
The preprocessed data will be saved to file `./data.json`.

### 6.7.2 Function Documentation

#### 6.7.2.1 def test\_indexer.main ( )

## 6.8 test\_speech Namespace Reference

### Functions

- def `main` ()

### 6.8.1 Detailed Description

This is to test the speech recognition functionality.

### 6.8.2 Function Documentation

#### 6.8.2.1 def test\_speech.main ( )

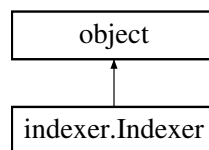


## Chapter 7

# Class Documentation

### 7.1 indexer.Indexer Class Reference

Inheritance diagram for `indexer.Indexer`:



#### Public Member Functions

- `def __init__ (self, source\_path, destination=None)`
- `def preprocess (self)`
- `def index (self)`

#### Public Attributes

- [source\\_path](#)
- [destination](#)
- [business](#)
- [review](#)
- [tip](#)
- [data](#)

#### 7.1.1 Detailed Description

This class is used to preprocess the yelp dataset and index the preprocessed data. The inverted index will be saved on disk.

## 7.1.2 Constructor & Destructor Documentation

### 7.1.2.1 `def indexer.Indexer.__init__( self, source_path, destination=None )`

`source_path`: the path to the yelp dataset folder  
`destination`: the destination file where to save the preprocessed data  
The business, review and tip data from the Yelp dataset are processed and indexed.  
The inverted index will be saved to folder `./index/`  
The preprocessed data will be saved to `"destination"` which can be read by searcher later on.

## 7.1.3 Member Function Documentation

### 7.1.3.1 `def indexer.Indexer.index( self )`

This function is used to index the preprocessed data.  
The inverted index will be saved to `./index/` folder  
`business_id`, `name`, `address`, `categories`, `review` and `tip` data are indexed.

### 7.1.3.2 `def indexer.Indexer.preprocess( self )`

This function is used to preprocess the yelp dataset.  
Data from the business, review and tip file will be combined,  
to form a single entity (a dict with `business_id` as key) that represents all the businesses.

## 7.1.4 Member Data Documentation

### 7.1.4.1 `indexer.Indexer.business`

### 7.1.4.2 `indexer.Indexer.data`

### 7.1.4.3 `indexer.Indexer.destination`

### 7.1.4.4 `indexer.Indexer.review`

### 7.1.4.5 `indexer.Indexer.source_path`

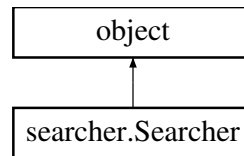
### 7.1.4.6 `indexer.Indexer.tip`

The documentation for this class was generated from the following file:

- `/Users/Federer-HYJ/Documents/EECS549/final project/YelpNLSearch/indexer.py`

## 7.2 searcher.Searcher Class Reference

Inheritance diagram for searcher.Searcher:



### Public Member Functions

- `def __init__ (self, path)`
- `def process_query (self, query, filters)`
- `def searching (self, query)`
- `def filtering (self, search_results, filters)`
- `def check_distance (self, id, value)`
- `def check_parking (self, id)`
- `def check_hours (self, id, time)`
- `def hour_to_number (self, hours)`

### Public Attributes

- `data`
- `analyzer`
- `reader`
- `searcher`

### 7.2.1 Constructor & Destructor Documentation

7.2.1.1 `def searcher.Searcher.__init__ ( self, path )`

### 7.2.2 Member Function Documentation

7.2.2.1 `def searcher.Searcher.check_distance ( self, id, value )`

check if a business resides in the requested distance radius

7.2.2.2 `def searcher.Searcher.check_hours ( self, id, time )`

function to check if a business opens at a given time

7.2.2.3 `def searcher.Searcher.check_parking ( self, id )`

function to check if a business has parking

#### 7.2.2.4 `def searcher.Searcher.filtering ( self, search_results, filters )`

Function to perform filtering.

#### 7.2.2.5 `def searcher.Searcher.hour_to_number ( self, hours )`

#### 7.2.2.6 `def searcher.Searcher.process_query ( self, query, filters )`

Function to process a query.

There are two steps: the first step uses the query to find relevant results ranked; the second step use the filter to eliminate items that do not match the filter.

#### 7.2.2.7 `def searcher.Searcher.searching ( self, query )`

Function to perform the search. Results will be returned based on relevance.

### 7.2.3 Member Data Documentation

#### 7.2.3.1 `searcher.Searcher.analyzer`

#### 7.2.3.2 `searcher.Searcher.data`

#### 7.2.3.3 `searcher.Searcher.reader`

#### 7.2.3.4 `searcher.Searcher.searcher`

The documentation for this class was generated from the following file:

- `/Users/Federer-HYJ/Documents/EECS549/final project/YelpNLSearch/searcher.py`

## Chapter 8

# File Documentation

### 8.1 /Users/Federer-HYJ/Documents/EECS549/final project/YelpNLSearch/distance.py File Reference

#### Namespaces

- [distance](#)

#### Functions

- def [distance.toRad](#) (degree)
- def [distance.distance](#) (p1, p2)

### 8.2 /Users/Federer-HYJ/Documents/EECS549/final project/YelpNLSearch/flask\_jqry.py File Reference

#### Namespaces

- [flask\\_jqry](#)

#### Functions

- def [flask\\_jqry.main](#) ()
- def [flask\\_jqry.view\\_do\\_something](#) ()
- def [flask\\_jqry.process](#) (q)

#### Variables

- [flask\\_jqry.app](#) = Flask(\_\_name\_\_)
- [flask\\_jqry.searcher](#) = None

### 8.3 /Users/Federer-HYJ/Documents/EECS549/final project/YelpNLSearch/indexer.py File Reference

#### Classes

- class [indexer.Indexer](#)

#### Namespaces

- [indexer](#)

### 8.4 /Users/Federer-HYJ/Documents/EECS549/final project/YelpNLSearch/query.py File Reference

#### Namespaces

- [query](#)

#### Functions

- def [query.all\\_cities](#) ()
- def [query.extract\\_entity\\_names](#) (t)
- def [query.parse\\_location](#) (original\_sentence)
- def [query.parse\\_time](#) (original\_sentence)
- def [query.parse\\_filters](#) (original\_sentence)
- def [query.query](#) (original)

#### Variables

- [query.times](#) = time.localtime()
- list [query.num](#)s
- dictionary [query.Cities](#) = {}
- dictionary [query.States](#) = {}
- dictionary [query.States\\_abbr](#) = {}
- float [query.laititude](#) = 36.169941
- float [query.longitude](#) = -115.139830

### 8.5 /Users/Federer-HYJ/Documents/EECS549/final project/YelpNLSearch/query\_speechrecognition.py File Reference

#### Namespaces

- [query\\_speechrecognition](#)



## Functions

- def `query_speechrecognition.all_cities` ()
- def `query_speechrecognition.get_audio_query` ()
- def `query_speechrecognition.extract_entity_names` (t)
- def `query_speechrecognition.parse_location` (original\_sentence)
- def `query_speechrecognition.parse_time` (original\_sentence)
- def `query_speechrecognition.parse_filters` (original\_sentence)
- def `query_speechrecognition.query_speech` ()

## Variables

- `query_speechrecognition.times` = `time.localtime()`
- list `query_speechrecognition.nums` = ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nigh', 'ten', 'eleven', 'twelve', 'thirtenn', 'fourteen', 'fifteen', 'sixteen', 'seventeen', 'eighteen', 'ninteen', 'twenty']
- dictionary `query_speechrecognition.Cities` = {}
- dictionary `query_speechrecognition.States` = {}
- dictionary `query_speechrecognition.States_abbr` = {}
- float `query_speechrecognition.laititude` = 36.169941
- float `query_speechrecognition.longitude` = -115.139830

## 8.6 /Users/Federer-HYJ/Documents/EECS549/final project/YelpNLSearch/README.md File Reference

## 8.7 /Users/Federer-HYJ/Documents/EECS549/final project/YelpNLSearch/searcher.py File Reference

### Classes

- class `searcher.Searcher`

### Namespaces

- `searcher`

## 8.8 /Users/Federer-HYJ/Documents/EECS549/final project/YelpNLSearch/static/js/bootstrap.min. js File Reference

### Functions

- if ("undefined"==typeof jQuery) throw new Error("Bootstrap requires jQuery")
- function `b` ()
- a fn `a` (function() {a.support.transition=`b`()}) (jQuery)
- a fn `a` fn `alert a` fn `alert c` prototype `close` (jQuery)
- `c` `hasClass` ("btn") || (c

## Variables

- function **a** {"use strict"
- **a** fn **emulateTransitionEnd** =function(**b**){var **c**=!1,**d**=this;**a**(this).one(**a**.support.transition.end,function(){**c**=!0});var **e**=function(){**c**||**a**(**d**).trigger(**a**.support.transition.end)};return setTimeout(**e**,**b**),this}
- var **b** ="[data-dismiss='alert']"
- var **c** =function(**c**){**a**(**c**).on("click",**b**,this.close)}
- **c** prototype **close** =function(**b**){function **c**(){f.trigger("closed.bs.alert").remove()}var **d**=**a**(this),**e**=**d**.attr("data-target");**e**||(e=**d**.attr("href"),e=e&&e.replace(/\.#(?=[^\s]\*\$)/,""));var **f**=**a**(**e**);**b**&&**b**.preventDefault(),**f**.length||(f=**d**.↵**hasClass**("alert")?**d**:**d**.parent()),f.trigger(**b**=**a**.Event("close.bs.alert")),b.isDefaultPrevented()||(f.remove↵**Class**("in"),**a**.support.transition&&**f**.**hasClass**("fade")?f.one(**a**.support.transition.end,**c**).**emulateTransition**↵**End**(150):**c**()})
- var **d** =**a**.fn.alert
- **a** fn **alert** =function(**b**){return this.each(function(){var **d**=**a**(this),**e**=**d**.data("bs.alert");**e**||**d**.data("bs.alert",e=new **c**(this)),**string**"==typeof **b**&&**e**[**b**].call(**d**)}}}
- **a** fn **a** fn **alert** **Constructor** =**c**
- **a** fn **a** fn **alert** **a** fn **alert** **noConflict** =function(){return **a**.fn.alert=**d**,this}
- **b** **DEFAULTS** ={loadingText:"loading..."}
- **b** **b** prototype **setState** =function(**a**){var **b** ="disabled",**c**=this.\$element,**d**=**c**.is("input")?"val":"html",**e**=**c**.↵**data**();**a**+="Text",**e**.resetText||**c**.data("resetText",**c**[**d**]()),**c**[**d**](**e**[**a**]||this.options[**a**]),setTimeout(function(){**loading**↵**Text**"==**a**?**c**.addClass(**b**).attr(**b**,**b**):**c**.removeClass(**b**).removeAttr(**b**),0)}
- **b** **b** prototype **b** prototype **toggle** =function(){var **a**=this.\$element.closest("[data-toggle='buttons']"),b=!0;if(**a**.↵**length**){var **c**=this.\$element.find("input");"radio"===**c**.prop("type")&&(c.prop("checked")&&this.\$element.↵**hasClass**("active")?b=!1:**a**.find(".active").removeClass("active")),b&&**c**.prop("checked",!this.\$element.has↵**Class**("active")).trigger("change")}b&&this.\$element.toggleClass("active")}
- **a** fn **button** =function(**c**){return this.each(function(){var **d**=**a**(this),**e**=**d**.data("bs.button"),f="object"==typeof **c**&&**c**;e||**d**.data("bs.button",e=new **b**(this,f)),**toggle**"==**c**?**e**.toggle():**c**&&**e**.**setState**(**c**)}}}

### 8.8.1 Function Documentation

8.8.1.1 **a** fn **a** ( **function**(){**a**.support.transition=**b**()} )

8.8.1.2 **function** **b** ( )

8.8.1.3 **a** fn **a** fn **alert** **a** fn **alert** **c** prototype **close** ( **jQuery** )

8.8.1.4 **c** **hasClass** ( "btn" )

8.8.1.5 **if** ( "undefined" ==typeof **jQuery** ) [new]

Bootstrap v3.0.3 (<http://getbootstrap.com>) Copyright 2013 Twitter, Inc. Licensed under [http↵://www.apache.org/licenses/LICENSE-2.0](http://www.apache.org/licenses/LICENSE-2.0)

## 8.8.2 Variable Documentation

8.8.2.1 a fn a fn button a fn button a {"use strict"

8.8.2.2 a fn alert =function(b){return this.each(function(){var d=a(this),e=d.data("bs.alert");e||d.data("bs.alert",e=new c(this)), "string"==typeof b&&e[b].call(d)}}}

8.8.2.3 function b ='[data-dismiss="alert"]'

8.8.2.4 a fn button =function(c){return this.each(function(){var d=a(this),e=d.data("bs.button"),f="object"==typeof c&&c;e||d.data("bs.button",e=new b(this,f)), "toggle"==c?e.toggle():c&&e.setState(c)}}}

8.8.2.5 var c =function(c){a(c).on("click",b,this.close)}

8.8.2.6 c prototype close =function(b){function c(){f.trigger("closed.bs.alert").remove()}var d=a(this),e=d.attr("data-target");e||(e=d.attr("href"),e=e&&e.replace(/.\*(?=#[^\s]\*\$)/,""));var f=a(e);b&&b.preventDefault(),f.length||(f=d.hasClass("alert"?d.parent():f.trigger(b=a.Event("close.bs.alert")),b.isDefaultPrevented()||(f.removeClass("in"),a.support.transition&&f.hasClass("fade"?f.one(a.support.transition.end,c).emulateTransitionEnd(150):c()))}

8.8.2.7 a fn a fn button Constructor =c

8.8.2.8 var d =a.fn.alert

8.8.2.9 b DEFAULTS ={loadingText:"loading..."}

8.8.2.10 a fn emulateTransitionEnd =function(b){var c=!1,d=this;a(this).one(a.support.transition.end,function(){c=!0});var e=function(){c||a(d).trigger(a.support.transition.end)};return setTimeout(e,b),this}

8.8.2.11 a fn a fn button a fn button noConflict =function(){return a.fn.alert=d,this}

8.8.2.12 b b prototype setState =function(a){var b="disabled",c=this.\$element,d=c.is("input"? "val": "html",e=c.data();a+="Text",e.resetText||c.data("resetText",c[d](e[a]||this.options[a]),setTimeout(function(){ "loading Text"==a?c.addClass(b).attr(b,b):c.removeClass(b).removeAttr(b)},0)}

8.8.2.13 b b prototype b prototype toggle =function(){var a=this.\$element.closest("[data-toggle="buttons"]"),b=!0;if(a.length){var c=this.\$element.find("input");"radio"===c.prop("type")&&(c.prop("checked")&&this.\$element.hasClass("active"?b=!1:a.find(".active").removeClass("active")),b&&c.prop("checked",!this.\$element.hasClass("active")).trigger("change")}b&&this.\$element.toggleClass("active")}

## 8.9 /Users/Federer-HYJ/Documents/EECS549/final project/YelpNLSearch/test\_indexer.py File Reference

### Namespaces

- [test\\_indexer](#)

## Functions

- def [test\\_indexer.main](#) ()

## 8.10 /Users/Federer-HYJ/Documents/EECS549/final project/YelpNLSearch/test\_speech.py File Reference

## Namespaces

- [test\\_speech](#)

## Functions

- def [test\\_speech.main](#) ()

# Index

/Users/Federer-HYJ/Documents/EECS549/final project/↔  
YelpNLSearch/README.md, [25](#)  
/Users/Federer-HYJ/Documents/EECS549/final project/↔  
YelpNLSearch/distance.py, [23](#)  
/Users/Federer-HYJ/Documents/EECS549/final project/↔  
YelpNLSearch/flask\_jqry.py, [23](#)  
/Users/Federer-HYJ/Documents/EECS549/final project/↔  
YelpNLSearch/indexer.py, [24](#)  
/Users/Federer-HYJ/Documents/EECS549/final project/↔  
YelpNLSearch/query.py, [24](#)  
/Users/Federer-HYJ/Documents/EECS549/final project/↔ c  
YelpNLSearch/query\_speechrecognition.py, [24](#)  
/Users/Federer-HYJ/Documents/EECS549/final project/↔  
YelpNLSearch/searcher.py, [25](#)  
/Users/Federer-HYJ/Documents/EECS549/final project/↔  
YelpNLSearch/static/js/bootstrap.min.js, [25](#)  
/Users/Federer-HYJ/Documents/EECS549/final project/↔  
YelpNLSearch/test\_indexer.py, [27](#)  
/Users/Federer-HYJ/Documents/EECS549/final project/↔  
YelpNLSearch/test\_speech.py, [28](#)  
\_\_init\_\_  
    indexer::Indexer, [20](#)  
    searcher::Searcher, [21](#)  
a  
    bootstrap.min.js, [26](#), [27](#)  
alert  
    bootstrap.min.js, [27](#)  
all\_cities  
    query, [13](#)  
    query\_speechrecognition, [15](#)  
analyzer  
    searcher::Searcher, [22](#)  
app  
    flask\_jqry, [12](#)  
b  
    bootstrap.min.js, [26](#), [27](#)  
bootstrap.min.js  
    a, [26](#), [27](#)  
    alert, [27](#)  
    b, [26](#), [27](#)  
    button, [27](#)  
    c, [27](#)  
    close, [26](#), [27](#)  
    Constructor, [27](#)  
    d, [27](#)  
    DEFAULTS, [27](#)  
    emulateTransitionEnd, [27](#)  
    hasClass, [26](#)  
    if, [26](#)  
    noConflict, [27](#)  
    setState, [27](#)  
    toggle, [27](#)  
    business  
        indexer::Indexer, [20](#)  
    button  
        bootstrap.min.js, [27](#)  
    c  
        bootstrap.min.js, [27](#)  
    check\_distance  
        searcher::Searcher, [21](#)  
    check\_hours  
        searcher::Searcher, [21](#)  
    check\_parking  
        searcher::Searcher, [21](#)  
    Cities  
        query, [14](#)  
        query\_speechrecognition, [16](#)  
    close  
        bootstrap.min.js, [26](#), [27](#)  
    Constructor  
        bootstrap.min.js, [27](#)  
    d  
        bootstrap.min.js, [27](#)  
    DEFAULTS  
        bootstrap.min.js, [27](#)  
    data  
        indexer::Indexer, [20](#)  
        searcher::Searcher, [22](#)  
    destination  
        indexer::Indexer, [20](#)  
    distance, [11](#)  
        distance, [11](#)  
        toRad, [11](#)  
    emulateTransitionEnd  
        bootstrap.min.js, [27](#)  
    extract\_entity\_names  
        query, [13](#)  
        query\_speechrecognition, [15](#)  
    filtering  
        searcher::Searcher, [21](#)  
    flask\_jqry, [11](#)  
        app, [12](#)  
        main, [12](#)

- process, 12
  - searcher, 12
  - view\_do\_something, 12
- get\_audio\_query
  - query\_speechrecognition, 15
- hasClass
  - bootstrap.min.js, 26
- hour\_to\_number
  - searcher::Searcher, 22
- if
  - bootstrap.min.js, 26
- index
  - indexer::Indexer, 20
- indexer, 12
- indexer.Indexer, 19
- indexer::Indexer
  - \_\_init\_\_, 20
  - business, 20
  - data, 20
  - destination, 20
  - index, 20
  - preprocess, 20
  - review, 20
  - source\_path, 20
  - tip, 20
- latitude
  - query, 14
  - query\_speechrecognition, 16
- longitude
  - query, 14
  - query\_speechrecognition, 16
- main
  - flask\_jqry, 12
  - test\_indexer, 17
  - test\_speech, 17
- noConflict
  - bootstrap.min.js, 27
- nums
  - query, 14
  - query\_speechrecognition, 16
- parse\_filters
  - query, 13
  - query\_speechrecognition, 15
- parse\_location
  - query, 13
  - query\_speechrecognition, 16
- parse\_time
  - query, 14
  - query\_speechrecognition, 16
- preprocess
  - indexer::Indexer, 20
- process
  - flask\_jqry, 12
- process\_query
  - searcher::Searcher, 22
- query, 12
  - all\_cities, 13
  - Cities, 14
  - extract\_entity\_names, 13
  - latitude, 14
  - longitude, 14
  - nums, 14
  - parse\_filters, 13
  - parse\_location, 13
  - parse\_time, 14
  - query, 14
  - States, 14
  - States\_abbr, 14
  - times, 14
- query\_speech
  - query\_speechrecognition, 16
- query\_speechrecognition, 14
  - all\_cities, 15
  - Cities, 16
  - extract\_entity\_names, 15
  - get\_audio\_query, 15
  - latitude, 16
  - longitude, 16
  - nums, 16
  - parse\_filters, 15
  - parse\_location, 16
  - parse\_time, 16
  - query\_speech, 16
  - States, 16
  - States\_abbr, 16
  - times, 16
- reader
  - searcher::Searcher, 22
- review
  - indexer::Indexer, 20
- searcher, 16
  - flask\_jqry, 12
  - searcher::Searcher, 22
- searcher.Searcher, 21
- searcher::Searcher
  - \_\_init\_\_, 21
  - analyzer, 22
  - check\_distance, 21
  - check\_hours, 21
  - check\_parking, 21
  - data, 22
  - filtering, 21
  - hour\_to\_number, 22
  - process\_query, 22
  - reader, 22
  - searcher, 22
  - searching, 22
- searching
  - searcher::Searcher, 22

- setState
  - bootstrap.min.js, [27](#)
- source\_path
  - indexer::Indexer, [20](#)
- States
  - query, [14](#)
  - query\_speechrecognition, [16](#)
- States\_abbr
  - query, [14](#)
  - query\_speechrecognition, [16](#)
- test\_indexer, [17](#)
  - main, [17](#)
- test\_speech, [17](#)
  - main, [17](#)
- times
  - query, [14](#)
  - query\_speechrecognition, [16](#)
- tip
  - indexer::Indexer, [20](#)
- toRad
  - distance, [11](#)
- toggle
  - bootstrap.min.js, [27](#)
- view\_do\_something
  - flask\_jqry, [12](#)