

---

# Flipgenic

**Daniel Thwaites**

**Apr 20, 2021**



# CONTENTS

<b>1 Documentation Contents</b>	<b>1</b>
1.1 Quickstart .....	1
1.2 flipgenic .....	2
<b>Python Module Index</b>	<b>5</b>
<b>Index</b>	<b>7</b>



## DOCUMENTATION CONTENTS

### 1.1 Quickstart

#### 1.1.1 Setup

1. Install Flipgenic from PyPI:

```
python -m pip install flipgenic
```

2. Download the spaCy model:

```
python -m spacy download en_core_web_md
```

3. Create an instance of `Responder`. This class will handle connection to the database, and provides methods for learning and recalling responses:

```
from flipgenic import Responder
responder = Responder('/path/to/storage/directory/')
```

#### 1.1.2 Learn Responses

Responses can be taught by passing a pair of statements, the second in response to the first:

```
responder.learn_response('Hello', 'Hi')
```

This is most commonly used in two ways:

- **Learn a large set of responses taken from a corpus.** This is helpful to build up the database when you first create a chatbot.
- **Learn each user input as a response to the last output in that conversation.** This extends your database as the bot is used, and adapts it to talk in a similar style to its users.

### 1.1.3 Get a Response

A response can be recalled as follows:

```
response, distance = responder.get_response('Hello')
```

`distance` is the distance between the vectors of the input text, and the text the selected response was originally responding to.

- The lower the value, the closer the match, and therefore the response has a higher chance of making sense in the conversation.
- If it is an exact match, it will be 0.
- The distance can theoretically be any positive number, however they tend to range from 0 up to around 5.
- If no responses are found (usually meaning the database is completely empty), the distance will be infinite.

By checking this value against a threshold of your choosing, you can opt not to output uncertain responses - or to replace them with a default message.

## 1.2 flipgenic

### 1.2.1 flipgenic package

```
class flipgenic.Responder (db_path, model='en_core_web_md')
```

Bases: object

Connects to database files and generates responses.

#### Parameters

- **db\_path** – Path to the database folder which will hold files related to this responder. Will be created if it doesn't exist.
- **model** – SpaCy model, or the name of one to be loaded.

```
add_response (responding_to, response)
```

Add a response to the batch, to be committed later.

`responding_to` is transformed into a vector now, but it is not added to the index until you commit it.

#### Parameters

- **response** – The response to be learned.
- **responding\_to** – The text this is in response to.

```
commit_responses ()
```

Ingest added responses into the database and index.

```
get_response (text)
```

Find the best response to the given input text.

**Parameters** `text` – Input text to respond to.

**Returns** Tuple of (response, distance).

```
learn_response (responding_to, response)
```

Learn the given text as the response to an input.

This is a shortcut to calling `add_response` followed by `commit_responses`, and therefore any other added responses will also be committed.

#### Parameters

- **response** – The response to be learned.
- **responding\_to** – The text this is in response to.

## Submodules

### flipgenic.db\_models module

```
class flipgenic.db_models.Response (**kwargs)
    Bases: sqlalchemy.orm.decl_api.Base
    id
    ngt_id
    response
```

### flipgenic.response module

```
flipgenic.response.get_closest_vector (text, index, nlp)
    Get the closest matching response from the index.
```

#### Parameters

- **text** – Text we are comparing against.
- **index** – NGT index to query from.
- **nlp** – Loaded SpaCy model for vectors.

**Returns** Tuple of (id, distance).

```
flipgenic.response.get_response (text, index, session, nlp)
    Generate a response to the given text.
```

#### Parameters

- **text** – Text to respond to.
- **index** – NGT index to use for queries.
- **session** – Database session to use for queries.
- **nlp** – Loaded SpaCy model for vectors.

**Returns** Tuple of (response, distance).

```
flipgenic.response.process_as_math (text)
    Attempt to process the text as a mathematical evaluation using mathparse.
```

**Returns** Response text, or None if the input cannot be parsed as math.

### flipgenic.vector module

`flipgenic.vector.average_vector` (*text*, *nlp*)

Get the vector for the given text.

This is calculated based on an average of SpaCy's word embeddings.

Ignore tokens which do not have a known vector, and punctuation. If this filtering removes all tokens, then fall back to SpaCy's implementation which includes everything.

#### Parameters

- **text** – Text string to process.
- **nlp** – Loaded SpaCy model for vectors.

**Returns** Average vector for the document.



## PYTHON MODULE INDEX

### f

flipgenic, 2  
flipgenic.db\_models, 3  
flipgenic.response, 3  
flipgenic.vector, 4



## A

add\_response() (*flipgenic.Responder* method), 2  
 average\_vector() (*in module flipgenic.vector*), 4

## C

commit\_responses() (*flipgenic.Responder*  
 method), 2

## F

flipgenic  
 module, 2  
 flipgenic.db\_models  
 module, 3  
 flipgenic.response  
 module, 3  
 flipgenic.vector  
 module, 4

## G

get\_closest\_vector() (*in module flip-  
 genic.response*), 3  
 get\_response() (*flipgenic.Responder* method), 2  
 get\_response() (*in module flipgenic.response*), 3

## I

id (*flipgenic.db\_models.Response* attribute), 3

## L

learn\_response() (*flipgenic.Responder* method), 2

## M

module  
 flipgenic, 2  
 flipgenic.db\_models, 3  
 flipgenic.response, 3  
 flipgenic.vector, 4

## N

ngt\_id (*flipgenic.db\_models.Response* attribute), 3

## P

process\_as\_math() (*in module flipgenic.response*),  
 3

## R

Responder (*class in flipgenic*), 2  
 Response (*class in flipgenic.db\_models*), 3  
 response (*flipgenic.db\_models.Response* attribute), 3