

# Midterm Report

FE595

Yubo Jin

## 1. Overview

When we search the information on Wikipedia, there are too many words on the web. We want to understand the definition in a short time. Is there a way to reduce the text and have the brief summary? We need to do something to summarize the text. In order to capture information more efficiently, I also extract the keywords, positive label, and negative labels.

In this project, I did something with natural language process using the python package 'nltk' and 'flask'.

## 2. Describe the documents

There are 5 documents in the 'Code' folder.

**main.py**: the main program to run

**midterm\_functions.py**: there are three functions for this project

**templates**: this is the folder containing the **midterm.html** file.

**positive-words.txt**: contain a lot of positive-words

**negative-words.txt**: contain a lot of negative-words.

## 3. Show the code step by step

(1) In the main.py.

```
import wikipedia
import midterm_functions as mf
from gensim.summarization.summarizer import summarize
from flask import Flask, render_template

app = Flask(__name__)

@app.route('/<name>', methods=['GET'])
def project(name):

    text = wikipedia.page(name)
    content = text.content
    summary = mf.get_summary(name)
    keywords = mf.get_keywords(content)
    abstract = summarize(content, ratio=0.05)
    pos_labels = mf.get_PosNegWords(abstract)[0]
    pos_labels = ', '.join(pos_labels)
    neg_labels = mf.get_PosNegWords(abstract)[1]
    neg_labels = ', '.join(neg_labels)

    return render_template('midterm.html', title = text.title,
                           summary = summary, keywords = keywords,
                           positive_words = pos_labels,
                           negative_words = neg_labels)

if __name__ == '__main__':
    app.run(host='127.0.0.1', port=8080, debug=True)
```

## (2) In the midterm\_functions.py

Import packages:

```
import nltk
import wikipedia
from nltk.stem.porter import PorterStemmer
from nltk.stem import WordNetLemmatizer
from gensim.summarization.summarizer import summarize
from gensim.summarization import keywords
```

Open the file:

```
# Open the file containing the positive words and negative words
with open('positive-words.txt', 'r') as file:
    positive_words = [line.strip() for line in file]
with open('negative-words.txt', 'r') as file:
    negative_words = [line.strip() for line in file]
```

Function 1:

```
# Get summary
def get_summary(name):
    text = wikipedia.page(name).content
    summary = summarize(text, ratio=0.03)
    return summary
```

Function 2:

```
# Get keywords
def get_keywords(text):
    key_words = nltk.word_tokenize(keywords(text, ratio=0.01))
    lmt = WordNetLemmatizer() # Lemmatizer
    porter = PorterStemmer() # Stemmer
    kw = [lmt.lemmatize(item) if lmt.lemmatize(item).endswith('e')
          else porter.stem(item) for item in key_words]
    kw = list(set(kw))

    # Find the words with same stemmer, keep the words end with 'e'
    duplicate = []
    for i in range(len(kw)):
        for j in range(i+1, len(kw)):
            if porter.stem(kw[i]) == porter.stem(kw[j]):
                a = kw[i]
                b = kw[j]
                for word in [a, b]:
                    if word.endswith('e'):
                        pass
                    else:
                        duplicate.append(word)
                if a.endswith('e') and b.endswith('e'):
                    duplicate.append(b)

    # Remove the words with duplicate stemmer, which doesn't end with 'e'
    kw = list(set(kw) - set(duplicate))
    return ', '.join(kw)
```

There are some drawbacks in the keywords function provided by 'nltk'. It only counts the frequency of each word except stop words, so there will be much duplicate keywords with same meaning such as learn, learns, learned, and learning. I remove the duplicate keywords

by stemming and lemmatization. We should also consider the situation that the words are end with letter 'e'. For example, when we stem the word 'include', the result is 'includ' so that it may cause some confuse to the user. I did something to fix it using the for loop.

### Function 3:

Get the positive words and negative words in the text.

```
# Get the negative labels and positive labels
def get_PosNegWords(text):

    # Not include all negations
    negations=['not', 'too', 'n\'t', 'no', 'cannot', 'neither','nor']
    tokens = nltk.word_tokenize(text)
    positive_tokens=[]
    negative_tokens=[]
    for i, token in enumerate(tokens):
        # When there is a positive word
        if token in positive_words:
            judger1 = True
            if i > 0:
                # A negation within N words?
                idx1 = i
                while judger1 is True:
                    idx1 = idx1 - 1
                    if idx1 == 0 or tokens[idx1] in ',.!? ' or tokens[idx1] == 'and':
                        break
                    elif tokens[idx1] in negations:
                        negative_tokens.append(tokens[idx1]+ ' ' + token)
                        judger1 = False
            if judger1 is True:
                positive_tokens.append(token)
        else:
            positive_tokens.append(token)

        # When there is a negative word
        elif token in negative_words:
            judger2 = True
            if i > 0:
                # A negation within N words?
                idx2 = i
                while judger2 is True:
                    idx2 = idx2 - 1
                    if idx2 == 0 or tokens[idx2] in ',.!? ' or tokens[idx2] == 'and':
                        break
                    elif tokens[idx2] in negations:
                        positive_tokens.append(tokens[idx2]+ ' ' + token)
                        judger2 = False
            if judger2 is True:
                negative_tokens.append(token)
        else:
            negative_tokens.append(token)

    return list(set(positive_tokens)), list(set(negative_tokens))
```

Next, we need to set some positive labels and negative labels for the text. There are a lot of positive words and negative words in positive-words.txt and negative-words.txt. If we find a word in the text is included in the positive word, does the meaning the word is positive? The answer is no. If there is a negation (no, not, can't) before the positive word, this 'positive word' should have the negative meaning. For instance, In the sentence 'I am not happy', we should classify the 'not happy' in the negative words instead of including in positive words.

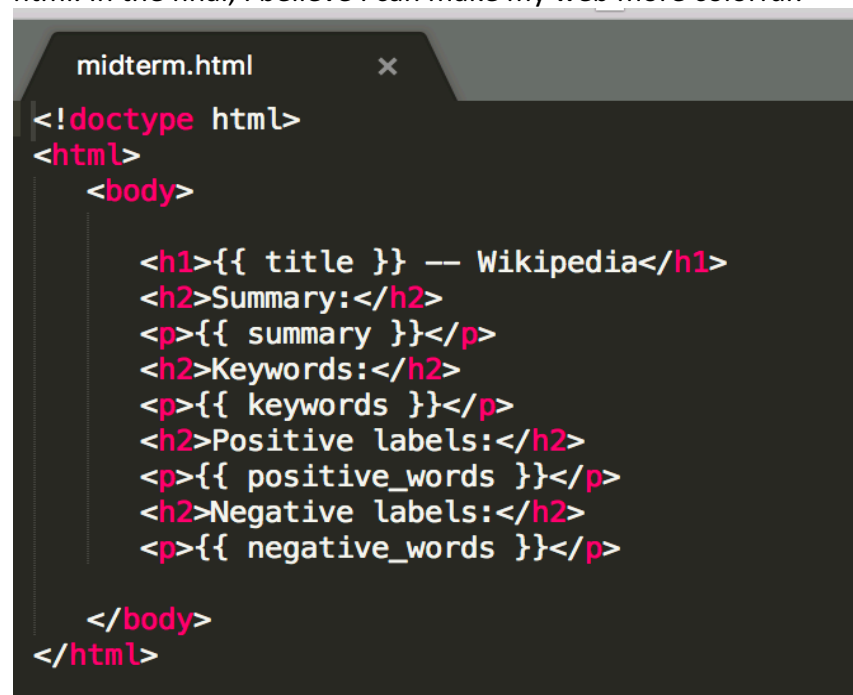
How to define the word's meaning is negative or positive?

- Negative sentiment:
  - negative words not preceded by a negation within n words in the same sentence.
  - positive words preceded by a negation within n words in the same sentence.
- Positive sentiment:
  - positive words not preceded by a negation within n words in the same sentence.
  - negative terms following a negation within n words in the same sentence.

We consider the words are negative sentiment and positive sentiment if the negation is within N words before the positive/negative words. It can be solved by testing backwards one by one from the positive/negative words to determine whether there is a word within negation until the beginning of this sentence.

### (3) In the midterm.html

I just made a simple web page structure in this document. I am learning how to code in html. In the final, I believe I can make my web more colorful!



```

midterm.html
<!doctype html>
<html>
  <body>

    <h1>{{ title }} -- Wikipedia</h1>
    <h2>Summary:</h2>
    <p>{{ summary }}</p>
    <h2>Keywords:</h2>
    <p>{{ keywords }}</p>
    <h2>Positive labels:</h2>
    <p>{{ positive_words }}</p>
    <h2>Negative labels:</h2>
    <p>{{ negative_words }}</p>

  </body>
</html>

```

## 4. testing result

Run the code in main.py

**Test 1:** Find the information about machine learning

Open the url [http://127.0.0.1:8080/machine\\_learning](http://127.0.0.1:8080/machine_learning) in the browser.

# Machine learning -- Wikipedia

**Summary:**

Machine learning (ML) is a field of artificial intelligence that uses statistical techniques to give computer systems the ability to "learn" (e.g., progressively improve performance on a specific task) from data, without being explicitly programmed.The name machine learning was coined in 1959 by Arthur Samuel. Machine learning is employed in a range of computing tasks where designing and programming explicit algorithms with good performance is difficult or infeasible; example applications include email filtering, detection of network intruders, and computer vision. Much of the confusion between these two research communities (which do often have separate conferences and separate journals, ECML PKDD being a major exception) comes from the basic assumptions they work with: in machine learning, performance is usually evaluated with respect to the ability to reproduce known knowledge, while in knowledge discovery and data mining (KDD) the key task is the discovery of previously unknown knowledge. The difference between the two fields arises from the goal of generalization: while optimization algorithms can minimize the loss on a training set, machine learning is concerned with minimizing the loss on unseen samples. Generalization in this context is the ability of a learning machine to perform accurately on new, unseen examples/tasks after having experienced a learning data set. Representation learning algorithms often attempt to preserve the information in their input but transform it in a way that makes it useful, often as a pre-processing step before performing classification or predictions, allowing reconstruction of the inputs coming from the unknown data generating distribution, while not being necessarily faithful for configurations that are implausible under that distribution.

**Keywords:**

machine, generalize, model, learn, data, algorithm

**Positive labels:**

intelligence, well, useful, work, proper, improve, good, accurately, not implausible, respect, leading

**Negative labels:**

difficult, problem, confusion, complex, not faithful, unknown, concerned, problems, bias, loss

## Test 2: Find the information about pasta

Open the url <http://127.0.0.1:8080/pasta> in the browser.

# Pasta -- Wikipedia

**Summary:**

Also commonly used to refer to the variety dishes made with it, pasta is typically made from an unleavened dough of a durum wheat flour mixed with water or eggs, and formed into sheets or various shapes, then cooked by boiling or baking. Some pastas can be made using rice flour in place of wheat flour to yield a different taste and texture, or for those who need to avoid products containing gluten.Pastas may be divided into two broad categories: dried (pasta secca) and fresh (pasta fresca). Common forms of pasta include long and short shapes, tubes, flat shapes or sheets, miniature shapes for soup, those meant to be filled or stuffed, and specialty or decorative shapes.As a category in Italian cuisine, both fresh and dried pastas are classically used in one of three kinds of prepared dishes: as pasta asciutta (or pastasciutta), cooked pasta is plated and served with a complementary side sauce or condiment; a second classification of pasta dishes is pasta in brodo, in which the pasta is part of a soup-type dish. However, the method of cooking these sheets of dough does not correspond to our modern definition of either a fresh or dry pasta product, which only had similar basic ingredients and perhaps the shape. Upon the addition of water, during mixing, intermolecular forces allow the protein to form a more ordered structure in preparation for cooking.Durum wheat is ground into semolina flour which is sorted by optical scanners and cleaned. The ingredients to make dried pasta usually include water and semolina flour; egg for colour and richness (in some types of pasta), and possibly vegetable juice (such as spinach, beet, tomato, carrot), herbs or spices for colour and flavour. Macaroni products – defined as the class of food prepared by drying formed units of dough made from semolina, durum flour, farina, flour, or any combination of those ingredients with water.

**Keywords:**

flour, cook, product, include, egg, pasta, sauce, fresh

**Positive labels:**

richness, complementary, flexible, variety, softer, fresh

**Negative labels:**

not fresh, boiling, not modern, lost

## Test 3: Find the information about panda

Open the url <http://127.0.0.1:8080/panda> in the browser.

# Giant panda -- Wikipedia

**Summary:**

The giant panda (*Ailuropoda melanoleuca*, literally "black and white cat-foot"; Chinese: 大熊猫; pinyin: dà xióng māo, literally "big bear cat"), also known as panda bear or simply panda, is a bear native to south central China. In captivity, they may receive honey, eggs, fish, yams, shrub leaves, oranges, or bananas along with specially prepared food.The giant panda lives in a few mountain ranges in central China, mainly in Sichuan, but also in neighbouring Shaanxi and Gansu. The giant panda has been referred to as a living fossil.Despite the shared name, habitat type, and diet, as well as a unique enlarged bone called the pseudo thumb (which helps them grip the bamboo shoots they eat) the giant panda and red panda are only distantly related. The giant panda is a "highly specialized" animal with "unique adaptations", and has lived in bamboo forests for millions of years. In captivity, zoos typically maintain the giant panda's bamboo diet, though some will provide specially formulated biscuits or other dietary supplements.Pandas will travel between different habitats if they need to, so they can get the nutrients that they need and to balance their diet for reproduction. Although adult giant pandas have few natural predators other than humans, young cubs are vulnerable to attacks by snow leopards, yellow-throated martens, eagles, feral dogs, and the Asian black bear. Giant panda cubs weigh 45 kg (100 pounds) at one year, and live with their mothers until they are 18 months to two years old.

**Keywords:**

bamboo, anim, zoo, bear, panda, china, giant, conserv, popul

**Positive labels:**

well

**Negative labels:**

wild, attacks, sloth, vulnerable, erroneously, limited