Aspect Based Emotion Analysis

Release 0.0.1

Quad-core

Dec 12, 2020

Aspect based emotion analysis aims to extract various aspects of reviews anddetermine the corresponding emotion for each aspect category. The term 'as-pect' refers to an attribute or a component of the product .Instead of classifyingthe emotion of overall review into anger, sadness, happiness, surprise and joyaspect-based analysis allows us to associate specific emotion to different aspectsof a product and such a analysis provides greater insight to the emotions ex-pressed in the written reviews.

Chapter 1

Aspect term Extraction

Aspect term extraction task is viewed as a sequence labelling problem.Eachtoken of review is marked with B,I,O encoding scheme.where B, I and O de-note the beginning,inside and outside entities of aspect terms.We have useda CRF(Conditional Random Field) model to classify the aspect terms. The classifier is trained with the following set of features:

1. Word information
2. Part-of-Speech (PoS) tag information
3. Previous chunk label information
4. Prefixes and suffixes

We were able to achieve an more than 80 percent accuracy.We have used nltklibrary for obtaining pos tag information.

Input: Not only was the food outstanding but the little perks were great

Output: ['O', 'O', 'O', 'O', 'B-A', 'O', 'O', 'O', 'O', 'B-A', 'O' 'O']

Aspects: ['food', 'perks']

Chapter 2

Emotion words Extraction

We have used stanford nlp for dependancy parsing and extract the dependencyrelations between the words in a sentence and extracted emotion related wordsfor each aspect in a sentence.

Input: Not only was the food outstanding but the little perks were great

Output: ['food': ['outstanding'], 'perks': ['little', 'great']]

Chapter 3

Aspect Emotion Detection

We tag all the aspects with a particular emotion based on the dependent words extracted. We have used 4 types of emotion tagging methods 1. NRC Lexicon 2. text2emotion library 3. Tf-idf SVM based 4.Logistic Regression

Input: [′food′: [′outstanding′], ′perks′: [′little′, ′great′]]

Output: [′food′: ′Happy′, ′perks′: ′Happy′]

Chapter 4

Mapping Emotion to polarity

Emotions Identified for each aspect were mapped as Positive , Negative and Neutral. 1. Trust, surprise, happy, joy - positive 2. Fear, anger, disgust, sadness - negative

Input: [’food’: ’Happy’, ’perks’: ’Happy’]

Output: [’food’: ’positive’, ’perks’: ’positive’]

Chapter 5

Main File

`Aspect.test.`**`Prec_rec`** (*Actual_polarity_dict_list, Predicted_polarity_dict_list*)

Extracting True positives, false positives and false negatives for calculation of precision and recall

Parameters
- **`Actual_polarity_dict_list`** (*`List`*) – List argument
- **`Predicted_polarity_dict_list`** (*`List`*) – List argument

`Aspect.test.`**`clean_data`** (*w*)

Clean the sentence by removing unwanted characters

Parameters **`w`** (*`String`*) – Sentence to be cleaned

Returns Cleaned sentence

Return type String

`Aspect.test.`**`crf_input`** (*input*)

Convert the input Sentence to feed into crf model for aspect extraction

Parameters **`input`** (*`String`*) – Input sentence

Returns List of features

Return type List of Dictionary

`Aspect.test.`**`dependencies`** (*txt, nlp*)

Extract the dependency relations from sentences

Parameters
- **`txt`** (*`String`*) – Input Sentence
- **`nlp`** (*`parser object`*) – Stanza parser object

Returns list of dependencies

Return type List of tuples

`Aspect.test.`**`dict_2_set`** (*dictionary*)

Function to convert the per sentence aspect based polarity dictionary into sets for ease in verification

Parameters **`dictionary`** (*`Dictionary`*) – Dictionary containg per aspect polarity.

Returns Dictionary converted to a set

Return type Set

`Aspect.test.`**`emotion_tagger`** (*emotion_words_per_aspect_dict*)

NRC lexicon based emotion tagger

Parameters **`emotion_words_per_aspect_dict`** (*`List of Dictionary`*) – Output Obtained from words function

Returns Two Lists. One contains emotions and other sentiment corresponding to each aspect

Return type Pair of list of Dictionary

`Aspect.test.`**`emotion_tagger2`** (*emotion_words_per_aspect_dict*)

text2emotion based emotion tagger

Parameters **`emotion_words_per_aspect_dict`** (*`List of Dictionary`*) – Output Obtained from words function

Returns Two Lists. One contains emotions and other sentiment corresponding to each aspect

Return type Pair of list of Dictionary

`Aspect.test.`**`emotion_tagger_SVM`** (*emotion_words_per_aspect_dict*)

SVM based emotion tagger

Parameters **`emotion_words_per_aspect_dict`** (*`List of Dictionary`*) – Output Obtained from words function

Returns Two Lists. One contains emotions and other sentiment corresponding to each aspect

Return type Pair of list of Dictionary

`Aspect.test.`**`emotion_tagger_lgr`** (*emotion_words_per_aspect_dict*)

Logistic Regression based emotion tagger

Parameters **`emotion_words_per_aspect_dict`** (*`List of Dictionary`*) – Output Obtained from words function

Returns Two Lists. One contains emotions and other sentiment corresponding to each aspect

Return type Pair of list of Dictionary

`Aspect.test.`**`emotions_counter`** (*Predicted_emotion_dict_list_tagger*)

Function to count the distriution of various emotions and plot a bar graph corresponding to each emotion tagger

Parameters **`Predicted_emotion_dict_list_tagger`** – List argument

Returns Counter object

`Aspect.test.`**`preprocess_and_tokenize`** (*data*)

Preprocessing data - cleaning

Parameters **`data`** (*`String`*) – Input Sentence

Returns A list of tokens

Return type List

`Aspect.test.`**`raw2dict`** (*sentence_nodes*)

Parse the xml file to extract ground truth values

Parameters **`sentence_nodes`** (*`Soup Object`*) – a soup object for parsing xml

Returns a list of dictionaries, contains id, text, aspect terms

Return type List of Dictionary

`Aspect.test.`**`sent2features`** (*sent*)
Convert a sentence to feature dictionary

Parameters **`sent`** (*`String`*) – Sentence to be converted

Returns List of feature dictionary

Return type List of Dictionary

`Aspect.test.`**`sent2features2`** (*sent*)
Convert a sentence to feature dictionary

Parameters **`sent`** (*`String`*) – Sentence to be converted

Returns List of feature dictionary

Return type List of Dictionary

`Aspect.test.`**`sent2labels`** (*sent*)
Convert a sentence to corresponding labels

Parameters **`sent`** (*`String`*) – Sentence for which labels are to be generated

Returns List of labels

Return type List

`Aspect.test.`**`word2features`** (*sent*, *i*)
Convert a word in a sentence to a feature dictionary

Parameters
- **`sent`** (*`String`*) – Sentence contaning the word to be converted to feature dict.
- **`i`** (*`Integer`*) – Index of the word in the sentence

Returns A dictionary object contaning features

Return type Dictionary

`Aspect.test.`**`words`** (*txt*, *featureList*, *nlp*)
Extract emotion related words from sentences

Parameters
- **`txt`** – Input sentence
- **`featureList`** (*`List`*) – List contaning aspects present in the sentence
- **`nlp`** (*`Parser object`*) – Stanza parser object

Returns List of aspects and corresponding emotion depicting words in the sentence

Return type List of Dicionary

Chapter 6

Indices and tables

- *Index*
- *Module Index*
- *Search Page*

Python Module Index

Index