ASPECT TERM EXTRACTION

CASE STUDY

Context

- Companies that provide services or products to their customers would like to know which **aspects** of their products preoccupy most of the customers.
- Are people talking the most about the **quality** of the food, the **price** of an item or the **battery life** of a computer?
- All those terms are what we call *aspect term*.

Some Background

The problem:

To identify aspect of product from a review, i.e. for each word of a review, identify which terms of their products preoccupy most of the customers.

Some terminologies:
The problem we are trying to solve can be broadly classified as Named Entity recognition

Definition:
Named Entities provides critical information for many NLP applications. Named Entity recognition and classification (NERC) in text is recognized as one of the important sub-tasks of Information Extraction (IE).

Available TOOLS

There are several ways to accomplish given task and following are three popular

1. NLTK & SCIKIT
2. Stanford's Named Entity Recognizer
3. Polyglot

POC uses approach 1.

NLTK Basics

Tagger:
A part-of-speech tagger, or **POS-tagger**, processes a sequence of words, and attaches a part of speech tag to each word

General N-Gram Tagging:
When we perform a language processing task based on unigrams, we are using one item of context.
This means we would tag a word such as *wind* with the same tag, regardless of whether it appears in the context *the wind* or *to wind*.
An **n-gram tagger** is a generalization of a unigram tagger whose context is the current word together with the part-of-speech tags of the *n*-1 preceding tokens.

Classification Algorithms:

Task: **Document/Text classification**

Document/Text classification is one of the important and typical task in *supervised* machine learning (ML). Assigning categories to documents, which can be a web page, library book, media articles, gallery etc. has many applications like e.g. spam filtering, email routing, sentiment analysis etc.

There are various algorithms which can be used for text classification.

Example: **Naive Bayes, Decision Tree, Maxent , Support Vector Machines (SVM) , Grid Search etc**

For POC there following were explored: **Naive Bayes**, **Decision Tree, Maxent.**

Classification Algorithms:

Task: **Document/Text classification**

Document/Text classification is one of the important and typical task in *supervised* machine learning (ML). Assigning categories to documents, which can be a web page, library book, media articles, gallery etc. has many applications like e.g. spam filtering, email routing, sentiment analysis etc.

There are various algorithms which can be used for text classification.

Example: **Naive Bayes, Decision Tree, Maxent , Support Vector Machines (SVM) , Grid Search etc**

For POC there following were explored: **Naive Bayes**, **Decision Tree, Maxent.**

Classification Algorithms

The metrics can be calculated by Precision and Recall Ideally:

There are four ways of being right or wrong:

- TN / True Negative: case was negative and predicted negative
- TP / True Positive: case was positive and predicted positive
- FN / False Negative: case was positive but predicted negative
- FP / False Positive: case was negative but predicted positive

For POC calculates only the accuracy of classifier:

1. accuracy NaiveBayesClassifier= 0.855523535062
2. accuracy DecisionTreeClassifier= 0.882612872238
3. accuracy MaxentClassifier= 0.871277617675

Things that can be done better

1. Using a better tagging and chunking method.
2. Evaluating several other algorithms
3. Cleaning data in a better way.
4. Exploring unsupervised aspect term methodologies to improve feature set.
5. Improvising in hyper parameters of algorithms used
6. Evaluation of metrics using Precision and Recall