USE CASE #4

PHILIP MORRIS INTERNATIONAL – DATA SCIENCE ASSESSMENT

BUSINESS UNDERSTANDING

- Companies that provide services or products to their customers would like to know which **aspects** of their products preoccupy most of the customers. This knowledge usually helps to improve product portfolio. 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*.
- You are asked to build an *Aspect Term Extractor* (ATE). An ATE is a model that extracts aspect terms from a review, i.e. for each word of a review, your model should predict if the word is an aspect term or not of the reviewed product.

DATA UNDERSTANDING (1)

- The input data has the following format: *review* → *list of aspect terms*, e.g. "The battery life is really good and its size is reasonable" → "battery life", "size". We recommend to change it and use the BIO format instead. Example:

The	battery	life	is	really	good	and	its	size	is	reasonable
O	B	I	O	O	O	O	O	B	O	O

- With this format, we can see that the role of an ATE is to assign to each word one of the three possible classes:
- O = not an aspect (Outside)
- B = first word of an aspect (Beginning)
- I = second, third, … word of an aspect (Inside)

DATA UNDERSTANDING (2)

- There are two data sets:-
  - *'Laptops_Train_v2.xml'* - to be used to train your model
  - *'Laptops_Test_Gold.xml'* - to be used to evaluate your model

DATA PREPARATION (1)

- Pre-processing is key for aspect parsing. The preprocessing module of the proposed framework consists of two major steps:
  - Firstly I extracted the nouns and adjectives in a sentence using the nltk POS Tagger and secondly
  - The sentence dependency tree is obtained through the xml.etree.ElementTree library

- XML Parser was created to extract root information for each entry. Example of each entry consist of the information below:-

```
<sentence id="892:1">
    <text>Boot time is super fast, around anywhere from 35 seconds to 1 minute.</text>
    <aspectTerms>
        <aspectTerm term="Boot time" polarity="positive"  from="0" to="9"/>
    </aspectTerms>
</sentence>
```

DATA PREPARATION (2)

- Aspect Term Extraction
  - Aspect term extraction is the key part of the assessment.
  - Quality of aspect terms detected will result in performances of other subtasks.
  - Words that follow a set of rules in Stanford Dependency Parser are filtered as aspect terms.
- To extract the terms defining quality of these aspect terms following approach is implemented:
  - Create a list of all adjective term in the sentence.
  - If there are no adjectives in a sentence than extract all the verbs in sentence after removing stop words in a list.
  - All words which are related to the aspect term up to a depth of 2 in Stanford Dependency Relation and present in the list are extracted as quality defining terms of aspect term.

FEATURE SELECTION – MOST COMMON ASPECTS (1)

- Identified 50 most common aspects:-
  - 'OS', 'Vista', 'Windows', 'Windows 7', 'applications', 'battery', 'battery life', 'charge', 'cost', 'design', 'display', 'extended warranty', 'features', 'games', 'gaming', 'graphics', 'hard drive', 'hardware', 'keyboard', 'keys', 'look', 'memory', 'motherboard', 'mouse', 'operating system', 'performance', 'power', 'power supply', 'price', 'processor', 'program', 'programs', 'quality', 'runs', 'screen', 'service', 'shipping', 'size', 'software', 'speakers', 'speed', 'system', 'use', 'value', 'warranty', 'warrenty', 'weight', 'windows', 'work', 'works'

- Used POS-Tagger
  - Gets POS tag for the sentence – which gives us part of speech tags for any sentence for all the words in the sentence

FEATURE SELECTION – MOST COMMON ASPECTS (2)

- The reviews are first segmented sentence-wise
- Used NLTK parser, to extract dependency relations between words in a sentence, and their Parts Of Speech tag
- From the dependency tree, the words are lemmatized
- Wrote certain rules based on the relations and POS tags, to extract the aspect terms
- A set of rules (noun, adjective, verb, adverb) were written, assuming the reviews (text) are grammatically correct
- Visualize aspect term in dataframe for test and training set
- Sort the dataframe pertaining to aspect's name

MODELING

- Before applying several modeling techniques, word vectors were generated using the CountVectorizer library
- Below were four algorithms that was applied to the datasets using the sklearn library:-
  - **Naive Bayes classifier for multinomial models(MultinomialNB),**
  - **Support Vector Classification (SVC),**
  - **Linear Support Vector Classification (LinearSVC),**
  - **Linear classifiers (SVM, logistic regression, a.o.) with stochastic gradient descent learning (SGDClassifier)**

EVALUATION (1)

- Aspect Term Detection
  - Number of Training Set Aspect Terms = 2358
  - Number of Test Set (Gold) Aspect Terms = 654

- MultinomialNB

Accuracy	Precision	Recall	F-Score
0.81970	0.88489	0.27765	0.42268

EVALUATION (2)

- SVC

Accuracy	Precision	Recall	F-Score
0.99671	0.99211	0.99435	0.99323

- LinearSVC

Accuracy	Precision	Recall	F-Score
0.99704	0.99212	0.99548	0.99380

EVALUATION (3)

- SGDClassifier

Accuracy	Precision	Recall	F-Score
0.99737	0.99324	0.99548	0.99436

CONVERTING TO BIOFORMAT

index	Text	BIO
0	The	O
1	battery	B
2	life	I
3	is	O
4	really	O
5	good	O
6	and	O
7	its	O
8	size	B
9	is	O
10	reasonable	O

CHALLENGES/ASSUMPTIONS

- Identifying aspects
  - Getting accurate aspects itself is a big challenge
  - Nouns may not always be aspect terms
- Errors in POS Tagger and NLTK Parser
- Using rule based method, however some of the aspect terms weren't extracted correctly by the rules

FUTURE WORK

- Discover more rules for aspect term extraction
- Combine existing rules for complex aspect extraction
- Incorporate aspect polarity to understand the sentiment of each review
- Incorporate aspect category detection
- To extract aspect categories, dictionaries created can be made more noise free
  - Detecting sarcasm and humors in the system