University of Sheffield

Aspect Based Sentiment Analysis

Harshad Gupta

Supervisor: Dr. Robert Gaizauskas

A report submitted in fulfilment of the requirements
for the degree of MSc in Computer Science with Speech and Language Processing

in the

Department of Computer Science

by

Harshad Gupta

September 12, 2018

Declaration

All sentences or passages quoted in this report from other people's work have been specifically acknowledged by clear cross-referencing to author, work and page(s). Any illustrations that are not the work of the author of this report have been used with the explicit permission of the originator and are specifically acknowledged. I understand that failure to do this amounts to plagiarism and will be considered grounds for failure in this project and the degree examination as a whole.

Name: Harshad Gupta

Signature:

Date: September 12, 2018

Abstract

Aspect Based Sentiment Analysis (ABSA) is the automated analysis of sentiments expressed by various reviewers towards different products and their aspects. This is required for decision making by businesses and organisations.
This project aims at developing and evaluating supervised learning techniques for the various subtasks of ABSA, viz. aspect classification and sentiment classification.
This project aims to develop a system to prove or disprove the hypothesis that "a system to solve the subtasks of SemEval 2016 ABSA would perform better when the feature to split the sentences which contains multiple ENTITY#ASPECT is implemented"

Acknowledgments

I am grateful to all my teachers,
who helped me in extracting the features,
on which my brain will train,
so that the knowledge that it gains will forever reign.

I take this opportunity to express my deepest gratitude to my supervisor Dr. Robert Gaizauskas, Professor in Computer Science, The University of Sheffield for his unwavering support and invaluable feedback. This project would not have been completed without his kind guidance.

I would also like to thank Dr. Andreas Vlachos for teaching the module of Natural Language Processing which laid the solid foundation required for this project.

And last but not the least, I would like to thank my parents and sister for their constant support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The advent of web 2.0 has seen the continued increase in user-generated content on the internet and its influence on the decision making of individuals and organisations [Ringsquandl and Petkovic, 2013]. In response to this, sentiment analysis and opinion mining of this content, which is the computational analysis of sentiments and opinions, has fast grown in importance [Pang and Lee, 2008]. Much of the research that has been done in the field of sentiment analysis and opinion mining is focused on computing the sentiment polarity of a whole review, which is of little use as it provides little or no information about the reviewer's opinion on particular aspects [Varghese and Jayasree, 2013]. Therefore, Aspect-Based Sentiment Analysis (ABSA), which tries to correctly identify each aspect, classify it with different entities, extract the linguistic expressions used to target the entity, and compute the sentiment polarity expressed for each aspect, is being researched [Álvarez-López et al., 2016]. In the opinionated text "Temple of Seitan serves some inexpensive yet delicious vegan chicken burgers", "vegan chicken burgers" is the linguistic expression used by the reviewer to target the entity "FOOD" and for whose "QUALITY" aspect the reviewer has expressed a "POSITIVE" sentiment.

1.1 Aims and Objectives

The aim of this project is to develop machine learning techniques for aspect identification and classification, and computing their sentiment polarity. For this aim to be achieved, the following objectives should be met:

- Obtain various annotated datasets for training.
- Develop and evaluate the best possible technique for aspect identification and classification and computing the sentiment polarity for each aspect.

1.2 Overview of the Report

This report is structured as follows:

- Chapter 2 is a survey of literature on the background information related to the project. In this chapter:

  - section 2.1 provides an overview of ABSA.
  - Section 2.2 provides an overview of machine learning;
  - Section 2.3 provides an overview of preprocessing;
  - Section 2.4 provides an overview of feature selection;
  - Section 2.5 provides an overview of summarization of the ABSA.
  - Section 2.6 provides an overview of SemEval.
  - Section 2.7 provides an overview of tools and resources available.
  - Section 2.8 provides a summary of the literature survey.

- Chapter 3 provides a detailed overview of analysis of the requirements and the evaluation of the project.
- Chapter 4 provides an overview about the design of the project.
- Chapter 5 provides an overview of the experiments performed and their results and the analysis of the results.
- Chapter 6 provides a brief conclusion of the project and their results and suggests some possible future works.

Chapter 2

Literature Survey

This chapter provides an overview of the previous research done and the literature on topics related to Aspect Based Sentiment Analysis (ABSA) and machine learning techniques focusing on supervised learning algorithms. Then, an overview of the methods for the summarization of the results, the Semantic Evaluation (SemEval) shared tasks, and the tools, that are available to carry out ABSA, are provided.

2.1 Aspect Based Sentiment Analysis

"Make No Bones is a nice place to eat. There are many vegan options, they all are delicious but a little expensive. Their doner kebab is to die for. Although it is far from the city center, it is always crowded."

Traditional sentiment analysis, which focuses on computing the overall sentiment of an opinionated text at the text or sentence level, will review the aforementioned opinionated text and comment that the reviewer has a positive sentiment towards the restaurant "Make No Bones". However, this type of sentiment analysis is unable to compute varying types of sentiments in the same context, regarding the same entity but possibly on different aspects of the said entity. For example, traditional sentiment analysis won't be able to comment that for the "style options", "quality", and "price" aspects of the entity "food" the reviewer has a positive, positive, and negative sentiment respectively. Traditional sentiment analysis also won't be able to tell that "doner kebab" was the linguistic expression used to target the entity "food" for whose "quality" aspect the reviewer has expressed a positive sentiment. Therefore, a more detailed and refined ABSA, which focuses on each aspect associated with a given entity individually; identifying it, extracting the linguistic expressions used to target its entity, and finding the leaning of its sentiment, is being researched [Toh and Su, 2016].

Understandably, ABSA refrains from taking opinion mining as a simple task of classifying sentiments. In its stead, ABSA introduces a range of problems requiring deeper Natural Language Processing (NLP) capabilities, which acts as a bridging tool between human (natural) language and various fields of on going research such as information extraction and machine learning. This also helps in deriving much deeper and richer results [Falk et al., 2016].

2.1.1 Tasks in Aspect Based Sentiment Analysis

Bing Liu defined the task of mining opinion at aspect level as the task of finding every quintuple (e_i, a_{ij}, oo_{ijkl}, h_k, t_l) in the given opinionated text, where "e" is the entity, "a" is the aspect of the entity, "oo" is the opinion on the aspect, "h" is the holder of the opinion, and "t" is the time at which opinion was made [Liu and Zhang, 2012]. The core tasks of ABSA are discussed in detail here:

1. **Aspect identification and classification:** The task is to identify the aspects about which the opinion has been made and classify them to a predefined set of entities like "FOOD" and "RESTAURANT". From the aforementioned opinionated text, this would be the extraction of aspect "style options", "quality", and "price" and then classifying them with the entity "food".

2. **Opinion Target Expression:** This task is actually a follow up of the first. Once the aspects have been identified and classified to the entities, the linguistic expressions which have been used to refer the reviewed entities in the opinionated text have to be extracted. In the aforementioned opinionated text, the linguistic expression "doner kebab" is used to refer the entity "food"[Pontiki et al., 2016].

3. **Aspect Sentiment Classification:** Once the aspects have been classified and the linguistic expressions targeting their entities have been extracted, these opinions then need to be classified to positive, negative, or neutral. In the aforementioned opinionated text, the opinion on the "price" aspect associated with the entity "food" is negative due to the opinion term "expensive" whereas the opinion on the "general" aspect of the entity "restaurant" is positive due to the opinion term "nice".

4. **Summarization:** Sentiment analysis is done on a huge number of reviews, each having many aspects. Therefore we need a way to summarize the opinions. The most common way of summarizing the opinions is on the basis of aspects, and can be done either quantitatively using bars and charts or qualitatively [Liu and Zhang, 2012].

The aforementioned first-three tasks are the subtasks of SemEval 2016 ABSA which have been discussed in detail in Section 2.7 and can be performed using different techniques of machine learning, and summarization can be performed quantitatively using graphs and charts. Machine learning is discussed in the following section and summarization in a later section.

2.2 Machine Learning

Machine learning is a set of techniques that for a given dataset helps the machine to learn the underlying patterns in the data which is then used to predict future data[Murphy, 2012]. Depending on the problem, machine learning can be classified into:

- **Supervised Learning** is the type of machine learning in which the machine learns to map the input to the output using the annotated training data provided. Classification and regression can be done using supervised learning [Murphy, 2012].

- **Unsupervised Learning** is the type of machine learning in which machine tries to learn the inherent structure of the data and group them according to their distinct features using the unlabelled training data. Clustering and association can be done using unsupervised learning [Floreano, 2008].

- **Reinforcement learning** is the type of machine learning that learns to augment the reward signals by associating situations with actions using a trial and error method [Sutton, 1992].

Supervised learning techniques are most suitable for this project as classification of aspects to the entities have to be done and annotated data is available. Supervised Learning is discussed in the following subsection.

2.2.1 Supervised Learning

Supervised learning is the process of adapting a system, in the presence of a learning base, which can confirm/condone the outcome for every input, so as to produce the desired outputs in response to given new inputs. Supervised learning actually is a kind of function approximation which modifies the function until it finally performs as expected [Reed and Marksll, 1999]. Supervised learning is done in two steps. In the first step, the system is trained using the annotated dataset (x_i, t_i) provided, which is a set of inputs x_i and their associated labels of target values t_i. The annotated training data, which will be used for this project, consists of a set of reviews and their target aspect and entity pairs and their associated sentiment polarity. For each review, the system will extract features and then map them to their target, adding a weight to the feature target pair each time the target matches with the predicted target for the concerned feature, and deducting a weight otherwise. In the end, the target which will have the greatest weight in the feature target pair will be mapped with the feature and stored in a model. In the second step, the system is provided with new reviews for which it is supposed to predict the target using the model created in the first step. The visual representation of supervised learning is provided in Figure 2.1 below. Classification, which is done using supervised learning, will be used for this project and is discussed in the following subsection.

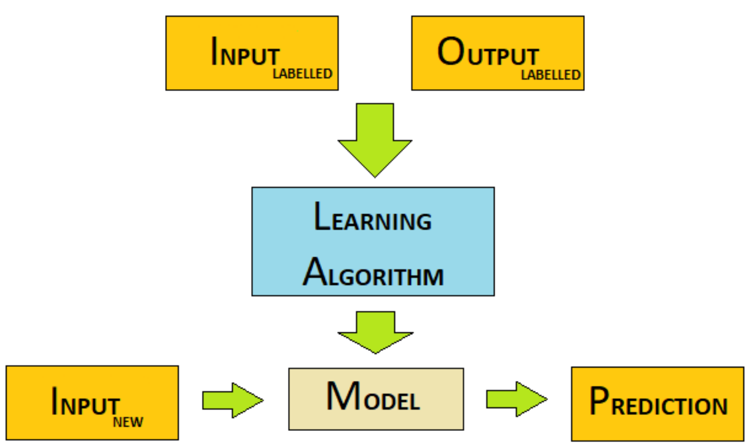

Figure 2.1: Supervised Learning

2.2.2 Classification

The task of classification needs labelled training data and hence it is done using supervised learning. Classification proceeds with creating a model using a set of manually annotated training data provided to the system with which new inputs are classified, on the basis of the features that are present in the input, to a set of classes which are predefined [Murphy, 2012]. For the classification of sentiments, features will be extracted from the reviews and will be mapped to the three predefined classes of sentiments, i.e. “neutral”, “positive”, and “negative”. After the training has been done and the model has been created, the features “delicious”, “nice”, and “expensive” will be classified to the class “positive”, “positive”, and “negative” respectively. Structured prediction, which is a kind of classification, is discussed in the following subsection.

2.2.3 Structured Prediction

Classification is the task of labelling an input with its corresponding target. Structured prediction involves the task of finding a sequence in addition to the task of classification. Named Entity Recognition (NER) is an example of structured prediction [Hofmann, 2009]. In the sentence, “Make No Bones is a nice place to eat”, the task of structured prediction is find the linguistic expression “Make No Bones”. It does so by classifying that “Make” and “Bones” as a part of linguistic expression, and then finding the sequence.

Different algorithms for the purpose of classification and structured prediction are discussed in the following subsection.

2.2.4 Algorithms

There are many Machine Learning algorithms for the tasks of classification, structured prediction, and clustering including Support Vector Machines, Convolutional Neural Networks, Naive Bayes, Conditional Random Fields, K-Nearest Neighbours, K-Means [Schapire],[Vlachos, 2013]. Brief overviews of a few algorithms are as follows:

1. **Naive Bayes** is probabilistic supervised learning model based on Baye's Theorem. It was designed primarily for the data in which features are not dependent on each other. For the data with feature dependency, it makes an assumption that features are independent [Kirk, 2014] [Gaizauskas, 2017].

2. **Stochastic Gradient Descent (SGD)** is a gradient descent for the optimisation of objective function in which the objective function gradient is estimated from one randomly selected example [Kroiss, 2016] [Zhaowen et al., 2015] [Ng].

3. **Support Vector Machines (SVMs)** is a non-probabilistic supervised learning model designed primarily for binary classification, but can be extended for multi-class classification and regression [Murphy, 2012]. It does classification by constructing an optimum hyperplane which maximizes the separation of hyperplane from the data points [Berwick, 2003] [Vlachos, 2004]. In this project classification will be done using SVMs. The LIBSVM library will be used for this.

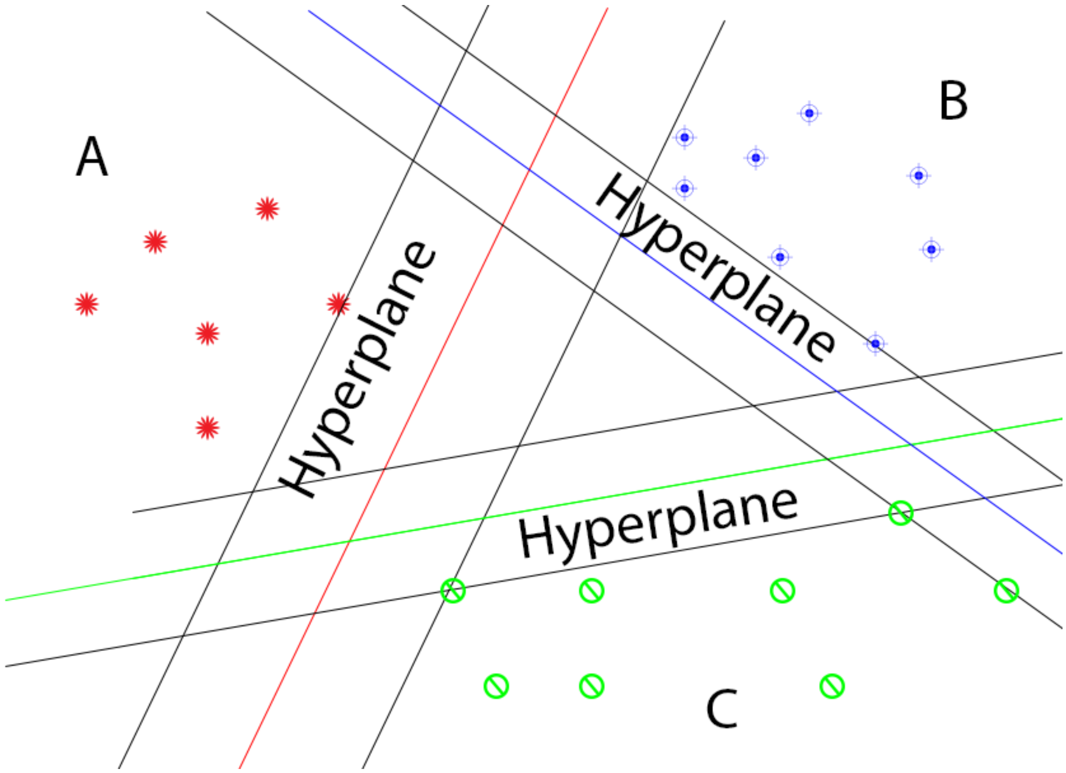

Figure 2.2: Maximum-margin hyperplane and margins for one-vs-all SVM

4. **Convolutional Neural Networks (CNNs)** is a kind of multilayered perceptron with deep and feed-forward neural networks which was initially developed for visual analysis. It is similar to Artificial Neural Networks (ANNs) but it requires reduced number of parameters [Savarese, 2018]. CNNs have three primary layers, which are "Convolutional layer", "Pooling layer", and "Fully-connected layer", and Input layer and Activation Layer. Input layer contains the input in the 3-Dimensional volume. The Convolutional layer is the main layer which extracts the features present in the input volume. The volume increases in the convolutional layer. Activation layer is mostly made up of Rectified Linear Unit (ReLU) activation function. Non-linear activation function like sigmoid and tanh can also be used, but ReLU is faster [**?**]. Volume does not change in the activation layer. Pooling layer downsamples the volume by discarding information. Fully-connected layer is the final learning layer which does the task of classification by passing the output vector through softmax [Li et al., 2018]. CNN can be used for aspect classification. The PyTorch library can be used for this.

5. **Conditional Random Fields (CRFs)** is a discriminative probabilistic undirected graphical model used for structured prediction which predicts labels for entities in the input sequences taking into context the features of the entities and those of their neighbours [Sutton and McCallum, 2011]. CRFs have many applications like **shallow parsing**, **Part-of-speech tagging**, **named entity recognition**, etc. [Murphy, 2012] CRFs canl be used for opinion target extraction. The PyStruct library can be used for this.

2.3 Preprocessing

Preprocessing of the training data is the first and crucial step for classification. Preprocessing of data includes cleaning and reduction of data which reduces redundant data from the whole training data, hence reducing the processing time and the memory use. This is achieved by doing tokenization, stop word removal, lowercase conversion, etc. [Xiang-wei and Yian-fang, 2012].

- **Tokenization** is the splitting of the text into smaller units, called tokens, while discarding certain unwanted characters like punctuation and spaces [Manning and Schiitze, 2018]. If the punctuation and spaces are required otherwise then the untokenized texts are saved as a copy. The input will be in form of sentences and the output will be in the form of a bag of words. Standford CoreNLP using NLTK can be used for tokenization.

- **Stop words removal** is the removal of certain words which are very common but are of no value for the document. For example "a", "as", "in", etc.

- **Lowercase conversion** is the conversion of every alphabetic character into lowercase. This reduces feature redudancy. For example, "The" and "the" will be "the".

- **Lemmatisation** is the removal of inflectional endings of words in order to get the base form of a word with the help of vocabulary and morphological analysis of words [Manning et al., 2008]. The lemmatisation of "ordered" and "ordering" is "order".

2.4 Feature Selection

For successfully performing machine learning using a supervised learning method, a huge amount of training data should be provided. The only drawback of a huge training dataset is that some features in the training data might be redundant. Teaching the system on these features is not only computationally extensive but also there is no improvement in the learning of the system using these features. Also, selecting better features, according to the given task, provides a better model of the data and helps the system to map the input to the output efficiently. Thus, it is important to do feature selection which is the selection of features that improves the mapping of input to the output. Some common feature selections are POS Tagging, Bigrams, Word Embeddings, Head Words, etc.[M. Dash, 1997].

- **POS Tagging** is the marking of words in a text corpus according to their Part-Of-Speech (POS) based on a word's relationship with related and adjacent words [Jurafsky and Martin, 20017]. POS tagging will help in aspect extraction, and will be done using NLTK.

- **Unigrams** are single words [Vlachos, 2018].

- **Bigrams** are pairs of consecutive words [Vlachos, 2018].

2.5 Aspect-Based Opinion Summary

ABSA is significant only if it is done on a huge volume of reviews as a small volume of reviews is not enough for any action. Therefore, there is a need for some form of summarization which is a crucial part of ABSA as it might save crucial amount of a reader's time which otherwise would be spent on reading all the reviews and filtering out the comments that are redundant [Fabbrizio et al., 2013]. The common way of summarization of sentiments is on the basis of aspects, and is known as *aspect-based opinion summary*. Summarization can be done either quantitatively or qualitatively [Liu and Zhang, 2012].

- **Quantitative summarization** is summarization in which the analysis of sentiments is visualized, and is crucial for many applications. This is done using different kinds of charts, viz. bar charts, pie charts, etc. This helps in visually comparing sentiments on multiple aspects. As the analysis is quantified, this type of summarization is more appropriate for analytical purposes [Liu and Zhang, 2012]. A quantitative summary may report that, "85 percentage of reviewers have positive sentiments towards 'Make No Bones', whereas 10 and 5 percentage of reviews hold negative and neutral sentiments respectively".

- **Qualitative summarization** is summarization in which a brief textual summary is produced. This provides an overview of people's sentiments towards different aspects to the reader. Qualitative summarization may report that, "Most people like 'Make No Bones'". The drawback of this type of summarization is it lacks any quantitative information, like how many people like "Make No Bones" [Liu and Zhang, 2012].

  Some opinionated texts are sentiment-laden in which similar sentiments are expressed multiple times. These are key measures in the overall polarity of the sentiments conveyed by the reviewer but are considered redundant by the techniques that are used in traditional summarization. Using qualitative summarization and a hybrid of abstractive and extractive summarization, a range of sentiments can be presented with the information about all the aspects of the concerned entity, instead of presenting just the facts and treating the whole topic as one [Fabbrizio et al., 2014].

2.6 Evaluation

In order to analyse how good a text classification system is, the effectiveness of the classifier in classifying the inputs to their corresponding targets have to be measured. This will be done by comparing the results of the system against a gold standard, which is an ideal result crafted by humans against which automated methods are compared. Standard evaluation metrics related to text classification will be used for this which are [Gaizauskas, 2017]:

1. **Accuracy**
   Accuracy of the system will be:

$$accuracy = \frac{\text{Number of correctly classified inputs}}{\text{Total number of inputs}}$$

   Systems that correctly classify an input with its associated class will achieve a score of 100% in system accuracy. While computing the accuracy is important, it does not always provide enough detail on how the system is performing for each individual class.

2. **Precision**
   Precision penalizes the system for classifying wrongly. Precision reports about the inputs that might have wrongly been classified as belonging to one class.

$$precision = \frac{\text{Number of inputs classified correctly}}{\text{Total number of inputs classified}} = \frac{\text{A}}{\text{A + B}}$$

3. **Recall**
   Recall reports how many inputs from the relevant class have been classified compared to all the inputs from the relevant class[Buckland and Gey, 1994].

$$recall = \frac{\text{Number of inputs classified correctly}}{\text{Total number of inputs that should have been classified correctly}} = \frac{\text{A}}{\text{A + C}}$$

4. **F-measure**
   F-measure reports the measure between precision and recall.

$$f - measure = \frac{\text{2 * Precision * Recall}}{\text{Precision + Recall}}$$

	Correct classification	Incorrect classification	Total
Classified (By System)	A (True Positive)	B (False Positive)	A + B
Not classified (By System)	C (True Negative)	D (False Negative)	C + D
Total	A + C	B + D	A + B + C + D

Table 2.1: Confusion Matrix [Stevenson, 2017]

2.7 SemEval

Semantic Evaluation (SemEval) is a series of international workshops for the "evaluations of computational semantic analysis systems", which is organized by the Special Interest Group on the Lexicon of the Association for Computational Linguistics (SIGLEX)[Sem]. The problem of ABSA, which is being solved in this project, was introduced as a task in SemEval 2014. The SemEval ABSA task was repeated in the 9th and the 10th workshop of SemEval in the year 2015 and 2016 respectively. The annotated training datasets, the evaluation measures, and the gold standard for the evaluation of different subtasks of SemEval ABSA was provided by SemEval. Datasets were provided for seven domains, viz., restaurant, laptop, hotel, camera, phone, etc. and eight languages, viz., English, Arabic, French, etc.

2.7.1 Subtasks of SemEval ABSA

The SemEval 2016 ABSA task is divided into three subtasks.

1. **Subtask 1: Sentence-level ABSA.** For a given opinionated text in which sentiment has been expressed about a target entity like a restaurant, a camera, etc., the goal of subtask 1 is to analyse the opinionated text at sentence level. For the opinionated text mentioned in section 2.1, the separate analysis of the sentence "Make No Bones is a nice place to eat" and the sentence "There are many vegan options, they all are delicious but a little expensive" will be a sentence-level analysis. Subtask 1 of SemEval ABSA task is divided into following three slots for which corresponding information has to be predicted.

   - **Slot 1: Aspect Category Detection.** For the slot 1 of the subtask 1 of the SemEval ABSA task, the goal is to predict, for every sentence of the opinionated text, pair of every entity E (e.g. Food, Restaurant, etc.) and attribute (aspect) A (e.g. Price, General, etc.) towards which the sentiment has been expressed. The entity and the attribute pair should be predicted from the list of predefined entity types and corresponding attribute labels. For the sentence "Make No Bones is a nice place to eat", the task is to predict "RESTAURANT#GENERAL" and for the sentence "There are many vegan options, they all are delicious but a little expensive", it is to predict "FOOD#STYLE_OPTIONS", "FOOD#QUALITY", and "FOOD#PRICE".
   - **Slot 2: Opinion Target Expression (OTE).** The goal of slot 2 is to extract the linguistic expression which has been used to refer to the reviewed entity of each entity attribute pair in the given sentence of the given opinionated text. If no entity has been explicitly mentioned then the value "NULL" should be returned. For the sentence "Make No Bones is a nice place to eat", "Make No Bones" should be extracted, and for the sentence "Although it is far from the city center, it is always crowded", the entity attribute pair is "LOCATION#GENERAL" but there is no linguistic expression that explicitly refers to the entity "LOCATION".
   - **Slot 3: Sentiment Polarity.** The goal of slot 3 is to gauge the polarity of the sentiment expressed in the sentence toward each identified entity aspect pair and assign either positive, negative, or neutral label to it. For the sentence "There are many vegan options, they all are delicious but a little expensive", the labels that will be assigned to the sentence are "POSITIVE", "POSITIVE", and "NEGATIVE" respectively.

   Subtask 1 has 7 different domains and 8 different languages.

2. **Subtask 2: Text-level ABSA.** For a given opinionated text in which sentiment has been expressed about a target entity like a restaurant, a camera, etc., the goal of subtask 2 is to analyse the complete opinionated text and summarize the sentiment expressed in the opinionated text. For the opinionated text mentioned in section 2.1, the complete text will be analysed and summarized. The analysis of the opinionated text for subtask 2 will be {RESTAURANT#GENERAL, NEUTRAL}, {'FOOD#STYLE_OPTIONS, POSITIVE}, {FOOD#QUALITY, POSITIVE}, {FOOD#PRICE, NEGATIVE}, and {LOCATION#GENERAL, NEGATIVE}. The sentiment polarity for entity aspect pair "RESTAURANT#GENERAL"

will be “NEUTRAL” because in the first sentence it is positive and in the last sentence it is “NEGATIVE”, so the polarities neutralises each other.

Subtask 2 has 3 different domains and 6 different languages.

3. **Subtask 3: Out-Of-Domain ABSA.** The goal of subtask 3 is to test the system for a domain for which training data was not made available.

   Subtask 3 has only the French language.

All the three subtasks have two modes, constrained and unconstrained. In the constrained mode the systems can be trained only with the training dataset which has been provided by SemEval, whereas in the unconstrained mode the system can be trained making use of any other resources (e.g. external lexica) and any other kind of additional data [Pontiki et al., 2016].

In SemEval 2016 ABSA, the term “attribute” has been used for the term “aspect”.

2.7.2 Datasets

Training, development, and gold standard datasets are provided by SemEval for training the system, parameter tuning, and evaluation of the system for each domain, language, and subtask except subtask 3 for which training and development datasets were not provided.

For subtask 1 of the Restaurant domain for the English language, the datasets consist of review text followed by corresponding pairs of entity and attribute, opinion target expression, the polarity of the sentiment, and the beginning and the ending index of the opinion target expression. The training dataset consists of 350 texts having 2000 sentences and 2507 entity attribute pairs. The testing dataset consists of 90 texts having 676 sentences and 859 entity attribute pairs [Pontiki et al., 2016].

```
<Review rid="15011990">
    <sentences>
        <sentence id="15011990:0">
            <text>Make No Bones is a nice place to eat.</text>
            <Opinions>
                <Opinion target="menu" category="RESTAURANT#GENERAL" polarity="positive" from="00" to="13"/>
            </Opinions>
        </sentence>
        <sentence id="15011990:1">
            <text>There are many vegan options, they all are delicious but a little expensive.</text>
            <Opinions>
                <Opinion target="vegan options" category="FOOD#STYLE_OPTIONS" polarity="positive" from="15" to="28"/>
                <Opinion target="vegan options" category="FOOD#QUALITY" polarity="positive" from="15" to="28"/>
                <Opinion target="vegan options" category="FOOD#PRICE" polarity="negative" from="15" to="28"/>
            </Opinions>
        </sentence>
    </sentences>
</Review>
```

Figure 2.3: An example of the review in the dataset

2.8 Tools and Resources

This section provides an overview of the tools that can be used for the project. These tools are for python 3 programming language as the project will be implemented using it.

2.8.1 NLTK

NLTK is Natural Language Toolkit built for python programming language for processing natural language data. Text processing operations like NER extraction, tokenization, POS Tagging, parsing, stemming, and classification can be done using the suite of libraries provided with it. Interfaces to resources such as WordNet, VerbNet, etc, and over 50 corpora are also provided with it [NLT].

2.8.2 WordNet

WordNet is a lexical database of English words, in which verbs, adverbs, noun, and adjectives are grouped into sets of synonyms, each of which expresses a distinct concept. WordNet is equivalent to a thesaurus, as it clusters words on the basis of their meanings [wor]. WordNet will be used for classification of aspects to the entities.

2.8.3 Stanford Parser

The **Stanford Parser** analyses sentences on the basis of their grammatical structures, does part-of-speech tagging, and identifies subjects and objects of verbs [sPa]. It will be accessed using the interface provided by NLTK, and will be used for POS Tagging.

2.8.4 LIBSVM

LIBSVM is an open source library for supervised machine learning using Support Vector Machine which was developed at National Taiwan University. It efficiently does multi-class classification, and will be used to do multi-class classification [lib].

2.8.5 PyTorch

PyTorch is a python version of Torch, an open source machine learning library for deep learning. For the building of neural networks it has a package called "nn" which supports both, feedforward and backpropogation [pyT]. It can be used to do multi-class classification using Convolutional Neural Networks.

2.8.6 PyStruct

PyStruct is an open source library for structured prediction. At the moment, only the max-margin methods and perceptron are implemented by it [pyS] [Meunier, 2017]. It can be used to do structured prediction using Conditional Random Fields (CRFs).

2.9 Summary

This chapter provided background information about ABSA, discussed machine learning and listed different types of it. Supervised learning was discussed along with the task of classification. Some of the algorithms regarding ABSA were introduced. The methods for summarization of results and their evaluation were discussed. Finally, the tools which will be utilized for the project were discussed briefly.

Chapter 3

Requirements and Analysis

This chapter provides an overview of the analysis of the problems that have been solved in the project and details of the requirements and the ethical, professional, and the legal issues associated with the project. The chapter begins with the analysis of the problems, followed by the discussion of the requirements and the evaluation of the system developed in the project. The chapter concludes with a discussion of the ethical, professional, and the legal issues associated with the project.

3.1 Problem Analysis

The project works on slots 1 and 3 of subtask 1 of the SemEval 2016 ABSA task for English language for the Restaurant domain. The English language was chosen because that is the only language in the options that I posses the knowledge of. Subtask 1 and the restaurant domain was chosen because the task of opinion target expression extraction was is only for subtask 1 and the restaurant domain. But, as a domain independent system is developed, the system was tested for slot 1 and slot 3 of subtask 1 for laptop domain.

In slot 1 the aspect category has to be identified. For the sentence "Make No Bones is a nice place to eat" in the opinionated text mentioned in section 2.1 the aspect "GENERAL" has to be identified and classified to the entity "RESTAURANT". Since the sentence has to be labelled with a label from a predefined set of labels, slot 1 is a multi-class classification problem. Using multi-class classification, the sentence will first be classified to an entity and then to an aspect making an entity aspect pair. In the aforementioned example, the sentence will be classified to the label "RESTAURANT#GENERAL".
Other option was to develop binary classifier for each separate entities and aspects, but it was rejected because the aim was to develop a domain independent system, and separate binary classifier would have made the system domain specific.

In slot 2 the opinion target expression has to be extracted. For the aforementioned sentence, the expression "Make No Bones" will be extracted. This is a structured prediction problem as not only the words have to be classified as a part of the expression but also the sequence of the words has to be identified. In the aforementioned sentence, "Make" and "Bones" will be classified as a part of the expression and then the sequence "Make No Bones" will be extracted [Pontiki et al., 2016].

In slot 3 the polarity of the sentiment has to be identified. For the aforementioned sentence, the polarity of the sentence has to be identified as “POSITIVE”. Since the sentence has to be labelled with a label from a predefined set of labels {“POSITIVE”, “NEGATIVE”, “NEUTRAL”}, slot 3 is also a multi-class classification problem.

Since a sentence can have multiple entity attribute pairs, the task also involves the problem of text simplification and splitting. In the sentence “There are many vegan options, they all are delicious but a little expensive.” there are three entity attribute pairs, viz., “FOOD#STYLE_OPTIONS”, “FOOD#QUALITY”, and “FOOD#PRICE”. The sentence has to be simplified first by copying the entity or the attribute or both from one part of the sentence to the other if that part lacks the entity or the attribute or both. For the aforementioned sentence, the simplified sentence will be “There are many vegan options. Vegan options all are delicious. Vegan options a little expensive.” After simplification the sentence will be split. The aforementioned sentence will finally become three sentences, viz., “There are many vegan options.”, “Vegan options all are delicious.”, and “Vegan options a little expensive.”.
The other option was to develop separate binary classifier for each entities and aspects, calculate the probability of each entity and aspect being classified for the text, and if the probability passes a threshold value then use that entity/aspect. This option was rejected because it required separate binary classifiers for each entities and aspects, and this would have made the system domain specific. The other reason because of which this option was rejected was that if multiple entity/aspect will exist in a sentence, each entity/aspect will read the signal from every other entity/aspect and the probability of each entity/aspect will be lesser and consequently the system won’t be able to learn that accurately.

3.2 Project Requirements

The main focus of this project is on the problem of simplification and classification of text. The primary goal of this project is to develop a domain independent system for splitting the sentences, identifying aspects, classifying them to their respective entity, extracting and mentioning the linguistic expression used for the entity, and classifying the sentiment expressed towards the aspects. The project tries to develop the best possible technique for these subtasks of ABSA and compares it with the performance of other teams that have taken part in the SemEval 2016 ABSA task. For this, the following are the requirements that have to be met:

1. **Collection of training datasets - Mandatory**

   The project will begin with the collection of annotated training sets, which will contain a huge amount of opinionated texts, from the SemEval website. Other open source opinionated training datasets from providers like Amazon and Yelp can also be used, but is not important for the successful completion of the project, as the ones provided by the SemEval will be enough, and these might only be used in order to improve the performance of the system.

2. **Preprocessing of datasets, splitting of sentences, and feature selection - Mandatory**

   Once the training datasets have been obtained, they will be preprocessed using different methods like tokenization, lemmatization, etc.. Sentences will be simplified and split, and different features will be selected.

3. **Development of the aspect and sentiment classifiers - Mandatory**

   The features selected will then be used to develop the classifier which will be used by the system using SVMs.

4. **Development of the structured prceptron - Optional**

   An algorithm for structured prediction will be developed using CRFs for the extraction of linguistic expression used to target the entities.

5. **Development of summarizer - Optional**

   A method for the summarization of opinions will be developed in order to summerize the analysis using quantitative summarizer. If time permits, a qualitative summarizer will also be worked on, and a method for doing information retrieval on the analysis will be developed. The development of summarizer is not important for the successful completion of the project, and it will only be an additional feature of the project.

6. **Evaluation - Mandatory**

   Once the system for text simplification, the classifiers, the algorithms for structure prediction, and the method for summarization have been developed, evaluation of these will be done using the gold standard provided by the SemEval.

3.3 Evaluation

To evaluate how good the system is at solving the problem of ABSA, the overall performance of the system has to be evaluated. This is done by measuring the effectiveness of different subtasks - splitting of the sentences, aspect classification, sentiment classification, and the extraction of linguistic expression used to refer the target entity. The evaluation will be done by comparing the results of the system against a gold standard. Gold standard is an ideal result crafted by humans against which automated methods are compared. This gold standard is provided by SemEval as the project works on the problem of ABSA proposed by it.

For aspect category detection and opinion target expression extraction (slot 1 and slot 2) F-1 score will be used as the measure for evaluation, and for splitting the sentences and sentiment polarity classification (slot 3) accuracy will be used as the measure for evaluation.

3.4 Ethical, Professional, and Legal Issues

There are no ethical or legal issues related to this project, as the dataset that is used by the system is provided by SemEval, a shared task project for research and development. All the tools and software used are either open source or a license has been received. The machine which is used is

provided by the University. Annotated training data were created by human participants, but the project does not involve any human participants. No confidential data is used.
Ethical review of the project is not sought as there are no ethical issues related to the project.

Chapter 4

Design

This chapter describes the design of the system which is implemented for ABSA. The chapter begins with an overview of the overall design of the system, followed by the overview of various parts, viz., preprocessing, sentence splitting, feature extraction, and the development of classifiers. The chapter concludes with the overview of the choice of the programming language that was made for the implementation of the system is given, followed by an overview of the tools used.

4.1 Overall System Design

The system has been divided into several parts. It begins with parsing the XML file containing the annotated corpus of text containing the opinionated text and the associated labels and opinion target expressions. This data is then preprocessed and a list of lemmatised words of the words that has been used in the opinion target expression is compiled and classified to predefined entities. Then, sentences are split using conjunctions and the classification of aforementioned compiled list of words. Then, the features are extracted from the text which are used to train the classifier and later for testing the classifier using the test dataset. These parts are shown in Figure 4.1 and in Figure 4.2 with an example below.

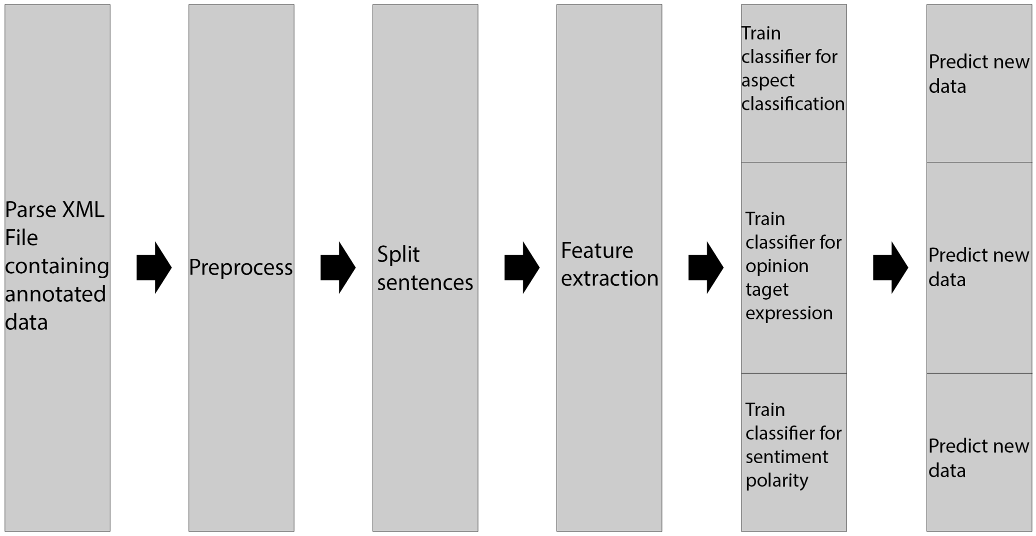

Figure 4.1: System Process

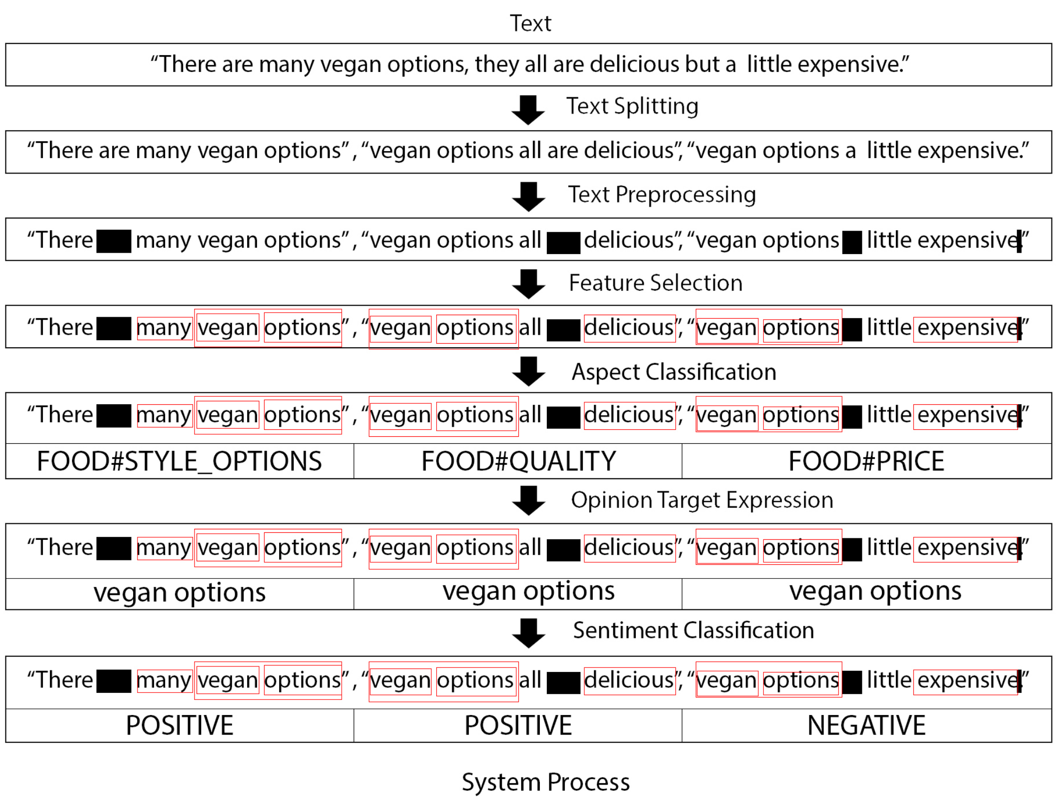

Figure 4.2: System Process with example

4.2 Preprocessing

The annotated corpus of text used for training and testing the classifier is in XML files. The data has to be preprocessed and converted into a format that can be read by the classifier. This is done by reading the XML file element by element, storing the opinionated texts in one list and the annotated targets in another. After this, the whole text is converted into lowercase and all the occurrence of currency symbols are substituted with the word “price” and the occurrence of the symbol “&” is substituted with the word “and”. From the text any word whose length is less that 3 characters is removed.

4.3 Sentence Simplification and Splitting

Sentences can have multiple entity attribute pairs. So, the sentences have to be simplified and split. For this, syntactic parsing and a rule-based approach has been used. First, a list of lemmatised words of the words that have been used in the opinion target expressions is compiled from the training dataset and these words are then classified to entities. Then, when a sentence is called for splitting, the sentence is tokenised and the lemmatised tokens are classified using SVM. If there exist pairs of tokens which are classified to entities and there exist splitting terms like “and”, “but”, “with”, “,”, “/”, “-”, the sentence is split at the position of these splitting terms. After this, dependency parsing is done on these split sentences and if there exist any conjunction then the sentence is split at the position of these conjunction. The sentence “Restaurant was nice, sushi was cheap and delicious.” will become “Restaurant-ENTITY was nice, sushi-ENTITY was cheap and delicious.” after entity classification. After this, splitting terms are searched for between the pairs of entities and “,” is found. “,” is substituted with the word “SPLIT”. After this, dependency parsing is done on the sentence and for every appearance of conjunctions, they are substituted with the word “SPLIT” and if no noun is in the one side of the word “SPLIT” then the noun from other side of the word “SPLIT” is copied to that side. So the sentence will become “Restaurant-ENTITY was niceSPLIT sushi-ENTITY was cheap SPLIT sushi-ENTITY delicious.”. From this “-ENTITY” is removed and the sentence is split at “SPLIT”. So finally the list of sentences after splitting is “Restaurant was nice”, “sushi was cheap”, and “sushi delicious”.

Algorithm 1 Pseudocode of text splitting

```
Do POS Tagging of the sentence
for each (word, POS) tupple in the sentence do
    get the lemma of the word
    classify the lemma to the entity
end for
for each splitting term do
    if splitting term between two lemma classified to entity do
        split at splitting term
    end if
end for
Do syntactic parsing of the sentence
for each element of the parsing do
    if element dependency is conjunction do
        split at conjunction
    end if
end for
```

4.4 Feature Extraction

After the preprocessing of the data and the splitting of the sentences, features are extracted from the opinionated texts which will be used to train and test the classifier. Following features have been extracted:

- **Unigrams** feature is extracted by removing the stopwords from the opinionated text and forming the bag of words.
- **Lemmatized Unigrams** feature is extracted by getting the lemma of the words of the text forming the bag of words of it.
- **Bigrams** feature is extracted by removing the stopwords from the text and forming the tuples of two consecutive words.
- **POS tags** feature is extracted by filtering the words based on its POS tag.

- **TFIDF** feature is extracted by using the "TfidfTransformer" method of "sklearn" library. TFIDF feature is the weight of the token depending on the term-frequency and the inverse of the document-frequency.

4.5 Development of Classifier

Various supervised machine learning algorithms have been studied in order to develop most accurate classifier possible for the subtasks. Multinomial Naive Bayes classifier (NB), Stochastic Gradient Descent classifier (SGD), and Linear Support Vector Classifier (LinearSVC) have been chosen for this project. Each of these classifiers has been trained using different features discussed in the previous section.

4.6 Programming Language

For this project Python programming language has been used. Python has been chosen because it is an open source language, easy to learn, and have many libraries for XML parsing, text processing, natural language processing, and machine learning.

4.7 Tools

For this project following Python libraries has been used:

- **xml.etree.ElementTree** has been used to parse the XML files which contains the datasets.
- **NLTK** has been used for processing natural language data. The following are the NLTK methods that have been used:
  - **pos_tag** has been used to get the Part-Of-Speech (POS) tag of each word.
  - **wordnet** has been used to get the wordnet POS which will be used for WordNetLemmatizer.
  - **WordNetLemmatizer** has been used to lemmatise a word depending on its POS tag.
  - **stopwords** is the NLTK corpus containing stopwords which has been used for stopword removal.
- **Spacy** has been used for dependency parsing for sentence splitting.
- **scikit-learn** has been used for feature extraction, development of classifier and its evaluation. Following are the scikit-learn methods that has been used:
  - **CountVectorizer** has been used to make feature vectors from the collection of opinionated text documents.
  - **DictVectorizer** has been used to make feature vectors from lists of mapping of feature and value.
  - **TfidfTransformer** has been used to convert the feature vectors into tf-idf representatives.
  - **LinearSVC** has been used to implement a Linear Support Vector Classifier.

- **MultinomialNB** has been used to implement a Multinomial Naive Bayes Classifier.
- **Pipeline** has been used to apply a list of transforms to the feature vectors in a sequence and then finally use it in the machine learning algorithm.
- **SGDClassifier** has been used to implement a linear classifier with Stochastic Gradient Descent (SGD) training.
- **metrics** has been used for the evaluation of the system using the F1_score and failure analysis using the classification_report and confusion_matrix

Chapter 5

Experiments and Results

This chapter gives the details of the experiments carried out and their results. Then, failure analysis and the final analysis of the system is done. The chapter concludes with the conclusion made from the result.

5.1 Experiments

5.1.1 Sentence Splitting

For splitting the sentences syntactic parsing and a rule-based approach has been used in following five different experiments:

1. **Syntactic Parsing only:** Splitting of the sentences is done only using the syntactic parsing at the occurrence of conjunctions.

2. **Syntactic Parsing and Word Classification:** Splitting of the sentences is done using the syntactic parsing and the word classification, and at the occurrence of conjunctions and the splitting terms between the two words that have been classified to entities.

3. **Syntactic Parsing followed by Word Classification:** Splitting of the sentences is done first using the syntactic parsing at the occurrence of conjunctions and then the word classification at the splitting terms between the two words that have been classified to entities.

4. **Word Classification only:** Splitting of the sentences is done only using the word classification at the splitting terms between the two words that have been classified to entities.

5. **Word Classification followed by Syntactic Parsing:** Splitting of the sentences is done first using the classification at the splitting terms between the two words that have been classified to entities followed by the syntactic parsing at the occurrence of conjunctions.

5.1.2 Feature Extraction

For feature extraction, following fourteen experiments have been performed:

1. **Unigrams:** For this experiment classifiers were trained only on the unigrams.

2. **Lowercase unigrams:** The classifiers used in this experiment were trained only on the lowercase unigrams.

3. **Preprocessed lowercase unigrams:** For this experiment classifiers were trained only on the preprocessed lowercase unigrams.

4. **Preprocessed lowercase unigrams with stopword removal:** In this experiment classifiers were trained only on the preprocessed lowercase unigrams with stopwords removed.

5. **Preprocessed lowercase lemmatised unigrams:** For this experiment classifiers were trained on the preprocessed lowercase unigrams and the lowercase lemmatised unigrams.

6. **Preprocessed lowercase lemmatised unigrams with stopword removal:** In this experiment classifiers were trained on the preprocessed lowercase unigrams with stopwords removed and the lowercase lemmatised unigrams.

7. **Preprocessed lowercase lemmatised unigrams with stopword removal and min. document frequency:** For this experiment classifiers were trained on the preprocessed lowercase unigrams with stopwords removed and the lowercase lemmatised unigrams whose minimum document frequency is 5.

8. **Preprocessed lowercase lemmatised unigrams and bigrams with stopword removal and min. document frequency syntactic parsing based splitting:** The classifiers used in this experiment were trained on the preprocessed lowercase unigrams and bigrams with stopwords removed and the lowercase lemmatised unigrams whose minimum document frequency is 5 after the sentences were split using only syntactic parsing based sentence splitting.

9. **Preprocessed lowercase lemmatised unigrams and bigrams with stopword removal and min. document frequency word classification based splitting:** For this experiment classifiers were trained on the preprocessed lowercase unigrams and bigrams with stopwords removed and the lowercase lemmatised unigrams whose minimum document frequency is 5 after the sentences were split using only word classification based sentence splitting.

10. **Preprocessed lowercase lemmatised unigrams with stopword removal and min. document frequency word classification and syntactic parsing based splitting and TFIDF:** For this experiment classifiers were trained on the preprocessed lowercase unigrams and their TFIDF with stopwords removed and the lowercase lemmatised unigrams whose minimum document frequency is 5 after the sentences were split using syntactic parsing based sentence splitting.

11. **Preprocessed lowercase lemmatised unigrams and bigrams with stopword removal and min. document frequency word classification and syntactic parsing based splitting and TFIDF:** For this experiment classifiers were trained on the preprocessed lowercase unigrams and bigrams with stopwords removed and the lowercase lemmatised unigrams whose minimum document frequency is 5 after the sentences were split using word classification and syntactic parsing based sentence splitting.

12. **Preprocessed lowercase lemmatised unigrams and bigrams with stopword removal and min. document frequency syntactic parsing followed by word classification based splitting and TFIDF:** For this experiment classifiers were trained on the preprocessed lowercase unigrams and bigrams and their TFIDF with stopwords removed and the lowercase lemmatised unigrams whose minimum document frequency is 5 after the sentences were split using syntactic parsing followed by word classification based sentence splitting.

13. **Preprocessed lowercase lemmatised unigrams and bigrams with stopword removal and min. document frequency word classification followed by syntactic parsing based splitting and TFIDF:** For this experiment classifiers were trained on the preprocessed lowercase unigrams and bigrams with stopwords removed and the lowercase lemmatised unigrams whose minimum document frequency is 5 after the sentences were split using word classification followed by syntactic parsing based sentence splitting.

14. **Preprocessed lowercase lemmatised unigrams and bigrams with stopword removal and min. document frequency word classification and syntactic parsing based splitting and TFIDF:** For this experiment classifiers were trained on the preprocessed lowercase unigrams and bigrams with stopwords removed and the lowercase lemmatised unigrams whose minimum document frequency is 5 for entity classifier and 6 for aspect classifier after the sentences were split using word classification and syntactic parsing based sentence splitting.

The use of the experiment described above have been documented in the sections 5.3.3 through section 5.3.6. The number shown in the figures in these sections are in tandem to the numbers mentioned in these experiments.

5.1.3 Classifier Development

For classifier development, three machine learning algorithms, Naive Bayes, SGD, and LinearSVC were experimented on similar features and for two algorithms, SGD and LinearSVC, two types of designs were experimented upon. Following are the five design of classifiers that have been experimented upon:

1. **Multinomail Naive Bayes Classifier:** A single Multinomail Naive Bayes classifier have been developed for the classification of ENTITY#ASPECT.

2. **Combined Stochastic Gradient Descent Classifier:** A single Stochastic Gradient Descent classifier have been developed for the classification of ENTITY#ASPECT.

3. **Separate Stochastic Gradient Descent Classifier:** Two separate Stochastic Gradient Descent classifier have been developed for the classification of ENTITY and ASPECT from the fifth experiment onward of feature selection. After the classification of ENTITY and ASPECT, decision function have been used to predict the ENTITY#ASPECT.

4. **Combined Linear Support Vector Classifier:** A single Linear Support Vector classifier have been developed for the classification of ENTITY#ASPECT.

5. **Separate Linear Support Vector Classifier:** Two separate Linear Support Vector classifier have been developed for the classification of ENTITY and ASPECT from the fifth experiment onward of feature selection. After the classification of ENTITY and ASPECT, decision function have been used to predict the ENTITY#ASPECT.

5.2 Measurements

Evaluation of a supervised machine learning algorithm can be done using several measures as described in section 3.3. For this project, F1-score and accuracy will be used as an evaluation measurement.

5.3 Results

Various experiments were conducted for the implementation of various parts of the system. The results of those experiments are as follows:

5.3.1 Word classification

Classification of words to entities were done using LinearSVC classifier for the purpose of splitting the sentences. The system was trained on 664 words and was tested for 884 words. The system achieved a F1-score of 87.019.

5.3.2 Sentence Splitting

Sentences were split because many sentences have more than one entity attribute pairs. Sentences were split using syntactic parsing and rule-based approach. Accuracy was the evaluation measurement, which calculates the total number of splits of the sentences that matches the number of entity attribute pairs in the sentences by total number of splits of the sentences. For the sentence "There are many vegan options, they all are delicious but a little expensive." there are three entity attribute pairs, viz., "FOOD#STYLE_OPTIONS", "FOOD#QUALITY", and "FOOD#PRICE". If the system splits the aforementioned sentence into three sentences then it is counted as accurate and if it splits the sentence in more than three or less than three sentences then it is not counted as accurate.

Following are the result of five experiments that were performed for splitting the sentences:

1. **Syntactic Parsing only:** Following was the performance of splitting system which used only the syntactic parsing:

	Equal	Lesser	Greater	Total	Accuracy
Training	1052	233	423	1708	61.59%
Testing	363	83	141	587	61.84%
Average	707.5	158	282	1147.5	61.72%

2. **Syntactic Parsing and Word Classification:** Following was the performance of splitting system which used both, the syntactic parsing and word classification:

	Equal	Lesser	Greater	Total	Accuracy
Training	1037	145	526	1708	60.71%
Testing	372	52	163	587	63.37%
Average	704.5	98.5	344.5	1147.5	62.04%

3. **Syntactic Parsing followed by Word Classification:** Following was the performance of splitting system which used the syntactic parsing followed by the word classification:

	Equal	Lesser	Greater	Total	Accuracy
Training	1129	145	434	1708	66.10%
Testing	392	52	143	587	66.78%
Average	760.5	98.5	288.5	1147.5	66.44%

4. **Word Classification only:** Following was the performance of splitting system which used only the word classification:

	Equal	Lesser	Greater	Total	Accuracy
Training	1191	245	272	1708	69.73%
Testing	421	88	78	587	71.72%
Average	806	166.5	175	1147.5	70.73%

5. **Word Classification followed by Syntactic Parsing:** Following was the performance of splitting system which used the word classification followed by the syntactic parsing:

	Equal	Lesser	Greater	Total	Accuracy
Training	1274	146	288	1708	74.59%
Testing	448	52	87	587	76.32%
Average	861	99	187.5	1147.5	75.46%

It has been observed that the splitting system which splits the sentence based on the word classification followed by the splitting based on syntactic parsing achieved the highest accuracy of 75.46% and thus performs the best. The Figure 5.1 below shows the performance of the system for different datasets, viz., training and testing, and the average of it for different implementation.

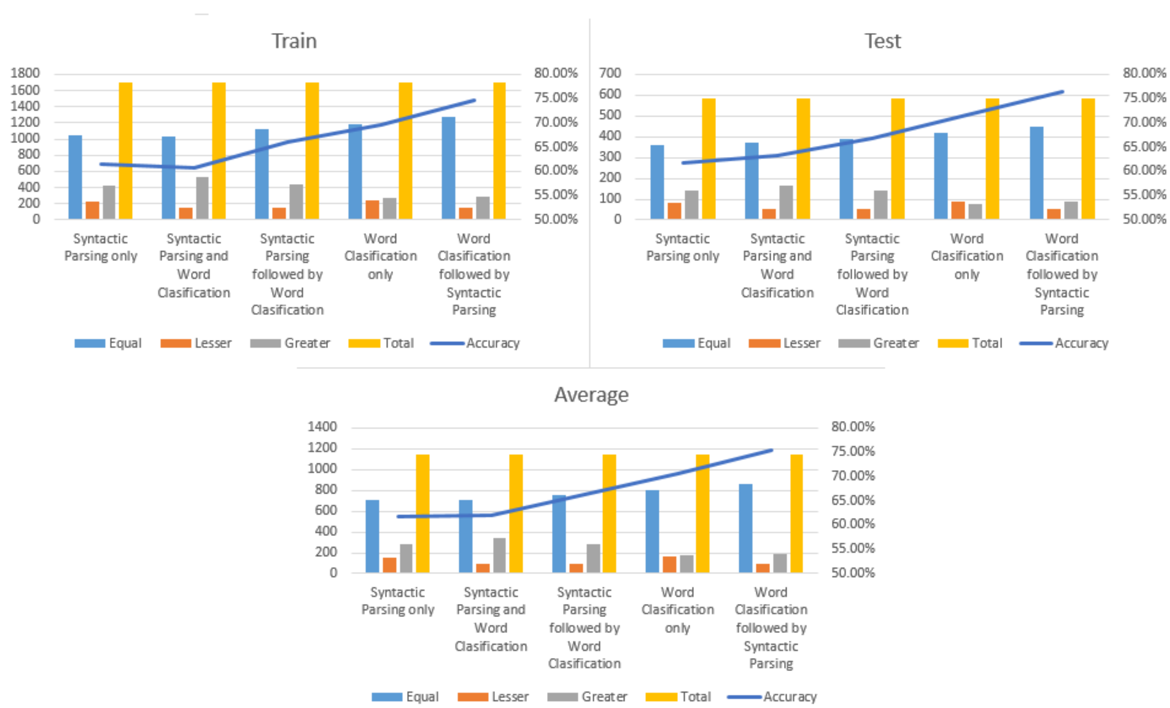

Figure 5.1: Splitting System Performance

5.3.3 Feature Extraction

A total of 14 combinations of feature extraction were experimented. For for Slot 1 (aspect classification) on linearSVC classifier a F1-score of 68.711 was achieved. Figure 5.2 and 5.3 shows the performance of different feature extraction combinations.

LinearSVC	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Entity#Aspect	59.932	60.048	60.419	60.203	60.274	61.036	60.83	64.502	63.844	64.856	65.722	66.12	65.763	65.722
Entity	69.494	69.494	70.377	71.023	70.206	70.865	70.851	75.546	75.326	77.344	78.424	77.944	78.204	78.424
Aspect	68.449	68.449	69.257	68.287	67.059	69.436	68.264	70.163	68.757	69.232	71.97	71.098	70.821	71.634
Polarity	77.411	77.411	77.269	75.259	76.354	76.897	77.89	80.103	79.749	79.906	79.471	79.348	79.581	79.488
Entity#Aspect	0	0	0	0	59.884	60.682	60.369	64.522	64.632	67.141	68.541	67.552	67.626	68.711

Figure 5.2: Feature Extraction Performance on LinearSVC classifier

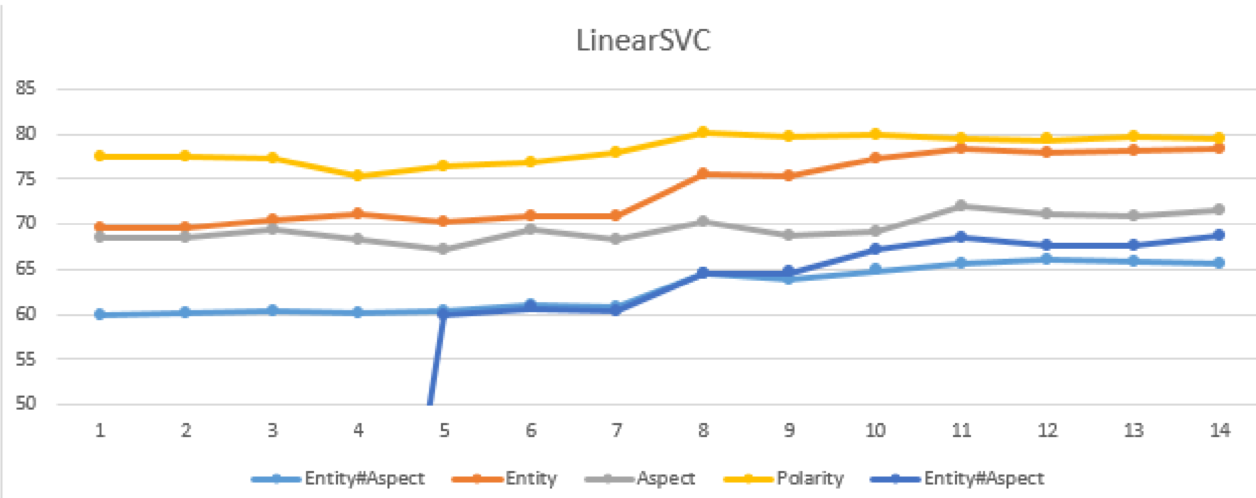

Figure 5.3: Feature Extraction Performance on LinearSVC classifier - Graph

5.3.4 Classifier Development

A total of 5 designs of classifiers were made. It was observed that among the three algorithms that were chosen, Multinomial Naive Bayes classifier performed the worst. Apart from three feature extraction, Stochastic Descent Gradient (SGD) classifier perform worse than the LinearSVC classifier, and apart from those three feature extraction, LinearSVC classifier performed the best. For SGD and LinearSVC classifier, two designs were made, one in which they were suppose to predict the ENTITY#ASPECT together and another in which they were supposed to predict ENTITY and ASPECT separately and then make an ENTITY#ASPECT pair. SGD classifier performed almost the same for both the designs, whereas the LinearSVC classifier better in the design in which it was supposed to predict ENTITY and ASPECT separately and then make an ENTITY#ASPECT pair. This is because for the separate prediction, it got more data to train on for lesser target label. Figure 5.4 below show the performance of different classifiers for different classifications and the comparison on the best design of the three algorithms on the ENTITY#ASPECT classification.

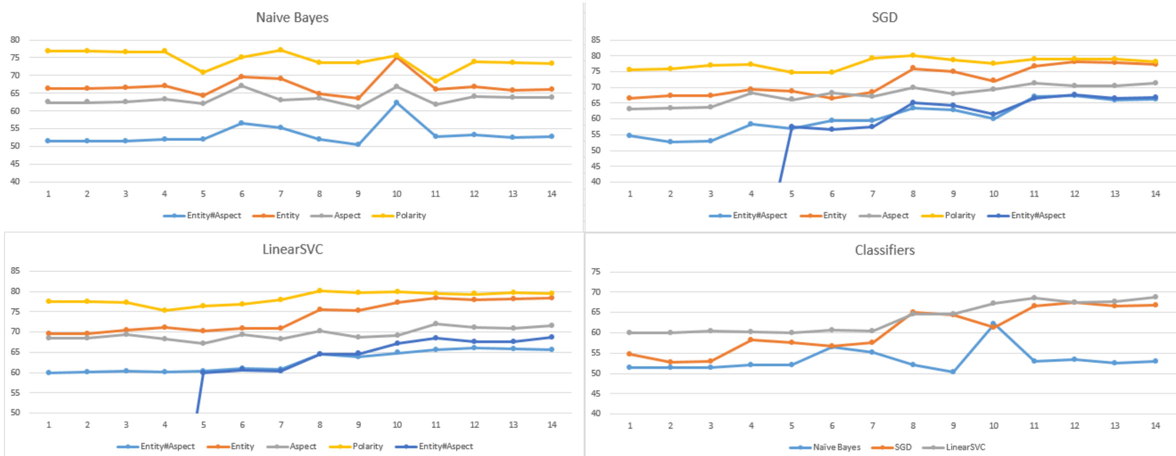

Figure 5.4: Classifier Comparison

5.3.5 Aspect Classification

The aspect classification was able the achieve the F1-score of 68.711. The Figure 5.5 shows the performance of aspect classification for different feature extraction for LinearSVC classifier.

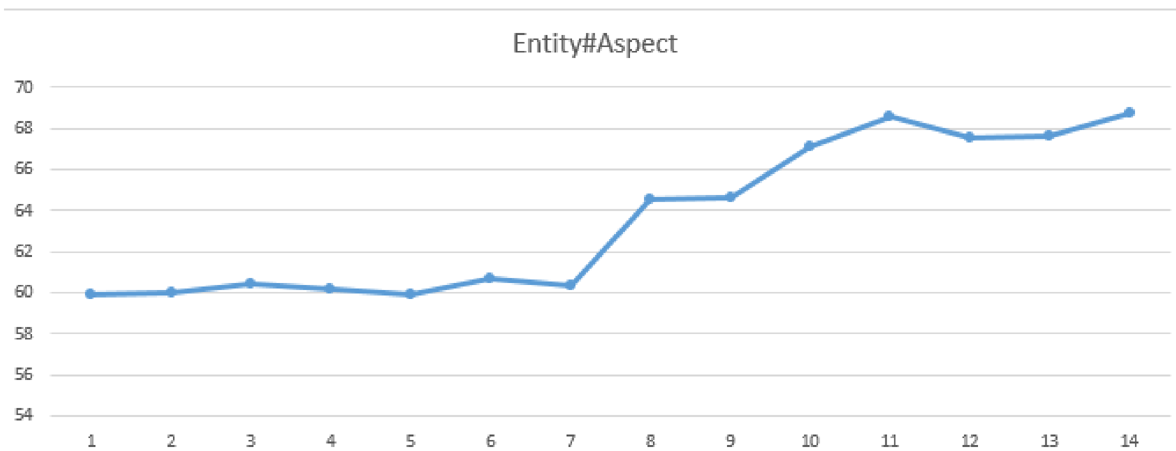

Figure 5.5: Entity#Aspect Classification Performance

5.3.6 Sentiment Classification

The sentiment classification was able to achieve the accuracy of 80.908%. The Figure 5.6 shows the performance of sentiment classification for different feature extraction for LinearSVC classifier.

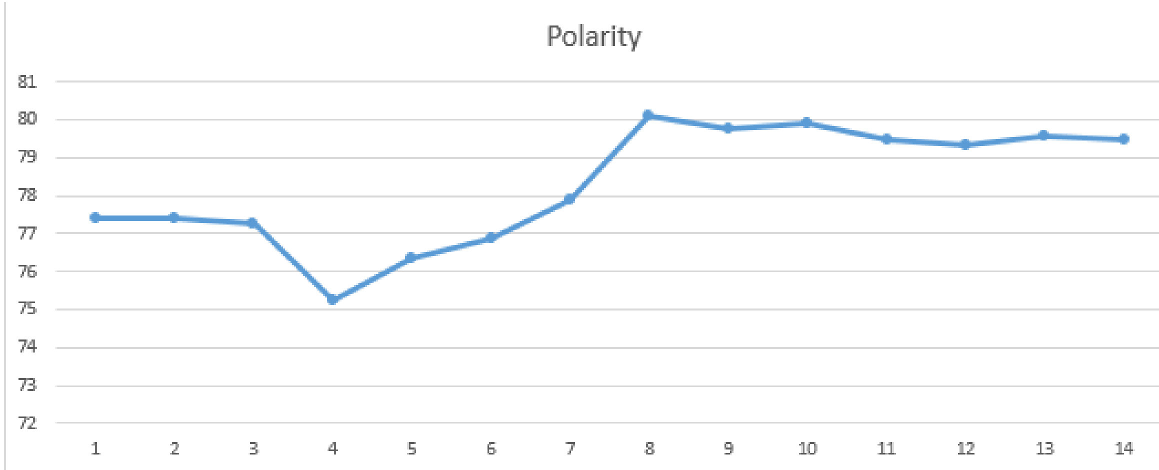

Figure 5.6: Polarity Classification Performance

5.3.7 Overall System Performance

The goal of the project was to develop a domain independent system, so although the system was developed keeping the restaurant domain in focus, the same unmodified system was tested for the laptop domain. The Aspect classification of the laptop domain was a much more fine grained task as compared to that of restaurant domain. This is because there are 71 ENTITY#ASPECT pairs in laptop domain as compared to restaurant domain which only have 12.

For slot 1 of the laptop domain, the system was able to pass the baseline F1-score of 37.481 and was able to achieve the F1-score of 42.176 in the constrained mode. The best system in constrained mode had the F1-score of 47.89 while the best system in unconstrained had the F1-score of 51.973 which the system was unable to achieve.

For slot 1 of the restaurant domain, the system performed well, and was able to achieve the F1-score of 68.711 which ranked it second among thirteen systems in the constrained mode and seventh among thirty one systems in the unconstrained mode. The best system in the constrained mode was able to achieve a F1-score of 71.494 and that in the unconstrained mode was able to achieve a F1-score of 73.031. Figure 5.7, 5.8, and 5.9 below shows the final performance of different classification, the standing of this system as compared to other systems in SemEval 2016 ABSA, and the performance of every system in SemEval 2016 ABSA and that of this system if this system would have taken part in it.

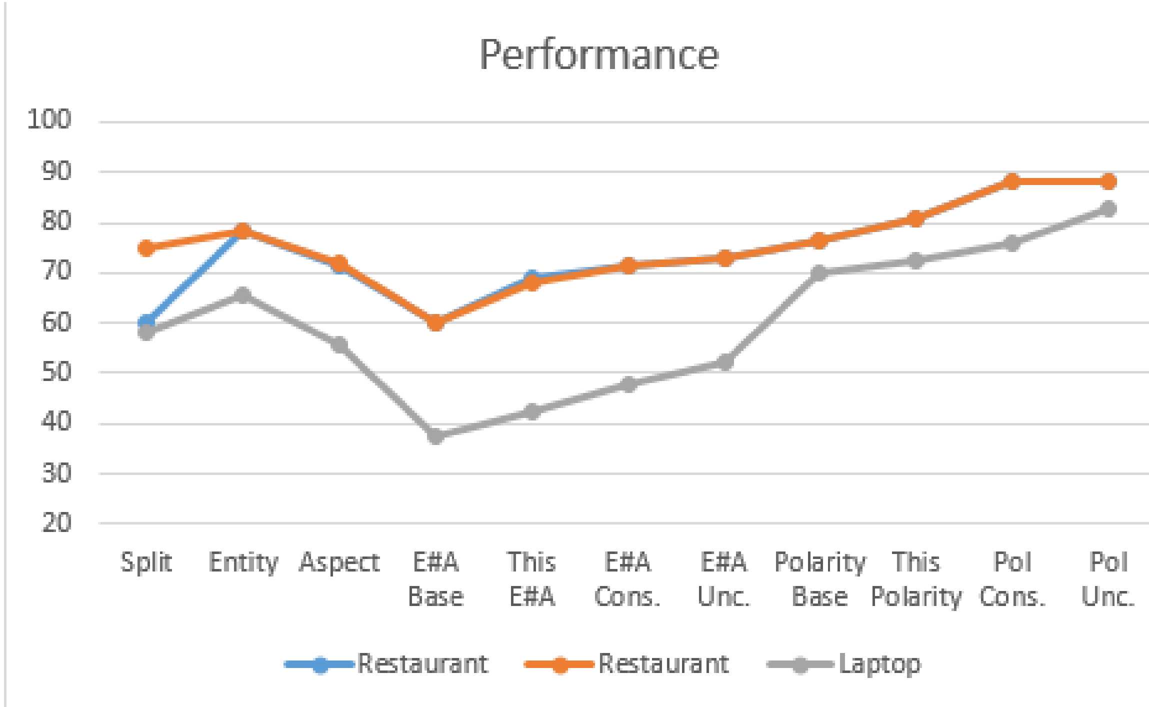

Figure 5.7: Overall System Performance

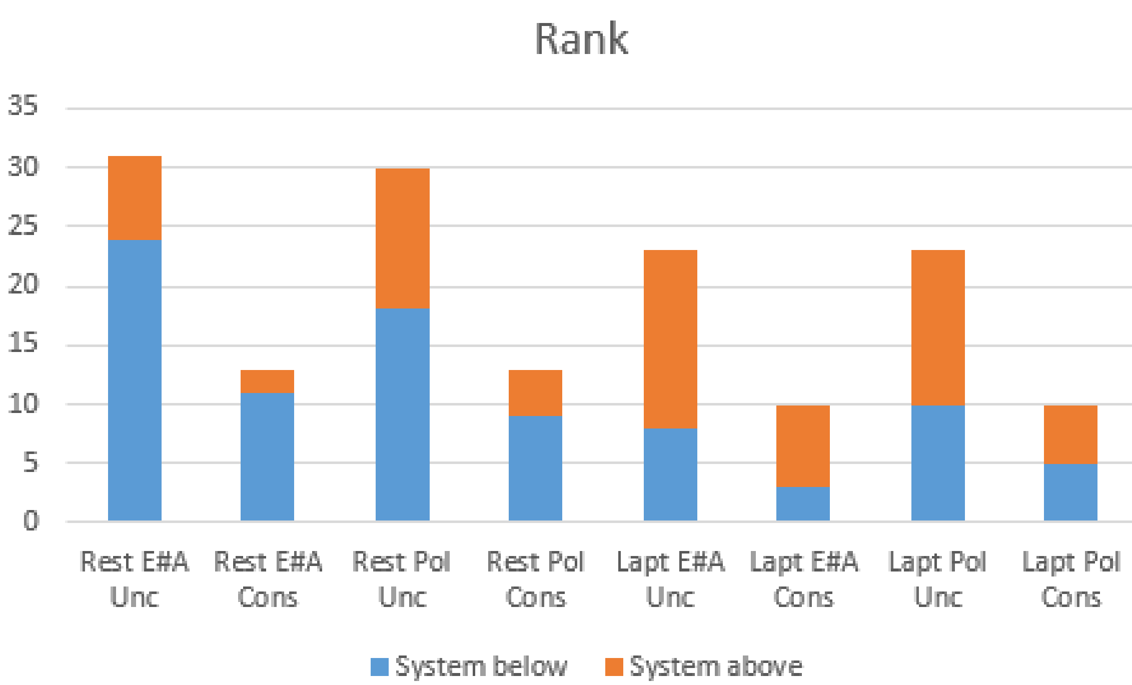

Figure 5.8: Rank of the system

NLANG./U/73.031	NLANG./U/72.34	XRCE/C/88.126
NileT./U/72.886	AUEB-./U/70.441	IIT-T./U/86.729
BUTkn./U/72.396	UWB/U/67.089	NileT./U/85.448
AUEB-./U/71.537	UWB/C/66.906	IHS-R./U/83.935
BUTkn./C/71.494	GTI/U/66.553	ECNU/U/83.586
SYSU/U/70.869	Senti./C/66.545	AUEB-./U/83.236
Shef/C/68.711	bunji/U/64.882	INSIG./U/82.072
XRCE/C/68.701	NLANG./C/63.861	UWB/C/81.839
UWB/U/68.203	DMIS/C/63.495	UWB/U/81.723
INSIG./U/68.108	XRCE/C/61.98	SeemGo/C/81.141
ESI/U/67.979	AUEB-./C/61.552	bunji/U/81.024
UWB/C/67.817	UWate./U/57.067	Shef/C/80.908
GTI/U/67.714	KnowC./U/56.816	TGB/C/80.908
AUEB-./C/67.35	TGB/C/55.054	ECNU/C/80.559
NLANG./C/65.563	BUAP/U/50.253	UWate./U/80.326
LeeHu./C/65.455	basel./C/44.071	INSIG./C/80.21
TGB/C/63.919*	IHS-R./U/43.808	DMIS/C/79.977
IIT-T./U/63.051	IIT-T./U/42.603	DMIS/U/79.627
DMIS/U/62.583	SeemGo/U/34.332	IHS-R./U/78.696
DMIS/C/61.754		Senti./U/78.114
IIT-T./C/61.227		LeeHu./C/78.114
bunji/U/60.145		basel./C/76.484
basel./C/59.928		bunji/C/76.251
UFAL/U/59.3		SeemGo/U/72.992
INSIG./C/58.303		AKTSKI/U/71.711
IHS-R./U/55.034		COMMI./C/70.547
IHS-R./U/53.149		SNLP/U/69.965
SeemGo/U/50.737		GTI/U/69.965
UWate./U/49.73		CENNL./C/63.912
CENNL./C/40.578		BUAP/U/60.885
BUAP/U/37.29		

Figure 5.9: Final performance of each system

5.4 Failure Analysis

The accuracy of the system is dependent on various factors. This includes, the vastness of training data available, proper splitting of the sentences, and the training of the system on the relevant features. If the system has not been trained in a particular feature, the system will be unable to predict new data pertaining to said feature thereby negatively impacting the accuracy of the system. An example of the effect of the system not being trained on the relevant feature can be found in the instance that suitable amount of training data was not available for the ENTITY#ASPECT DRINKS#PRICES label. This led to the under-performance of the system corresponding to this label. The results of the same can be seen in the confusion matrix shown in the Figure 5.11 below. There are sentences which the system is unable to split because either it does not contain any conjuction or it does not contain any previously defined spliting terms, but have more than one clauses. If it were possible to develop a feature that would help the system to split these types of sentences, the accuracy of the system would be enhanced further. For example, the system was unable to split the sentence "The food was delicious for such a cheap price" which contained FOOD#QUALITY and FOOD#PRICES ENTITY#ASPECT label.

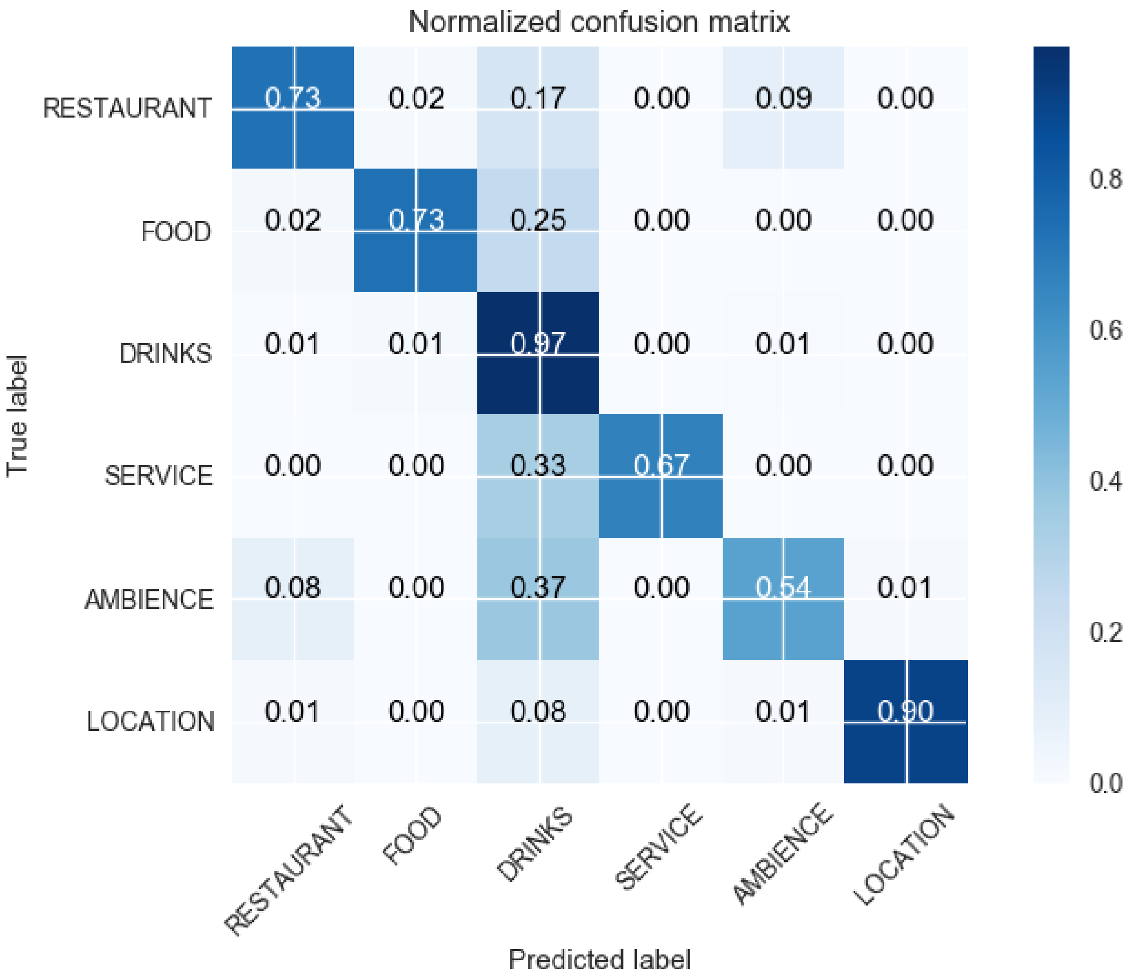

Figure 5.10: Confusion matrix for word classification

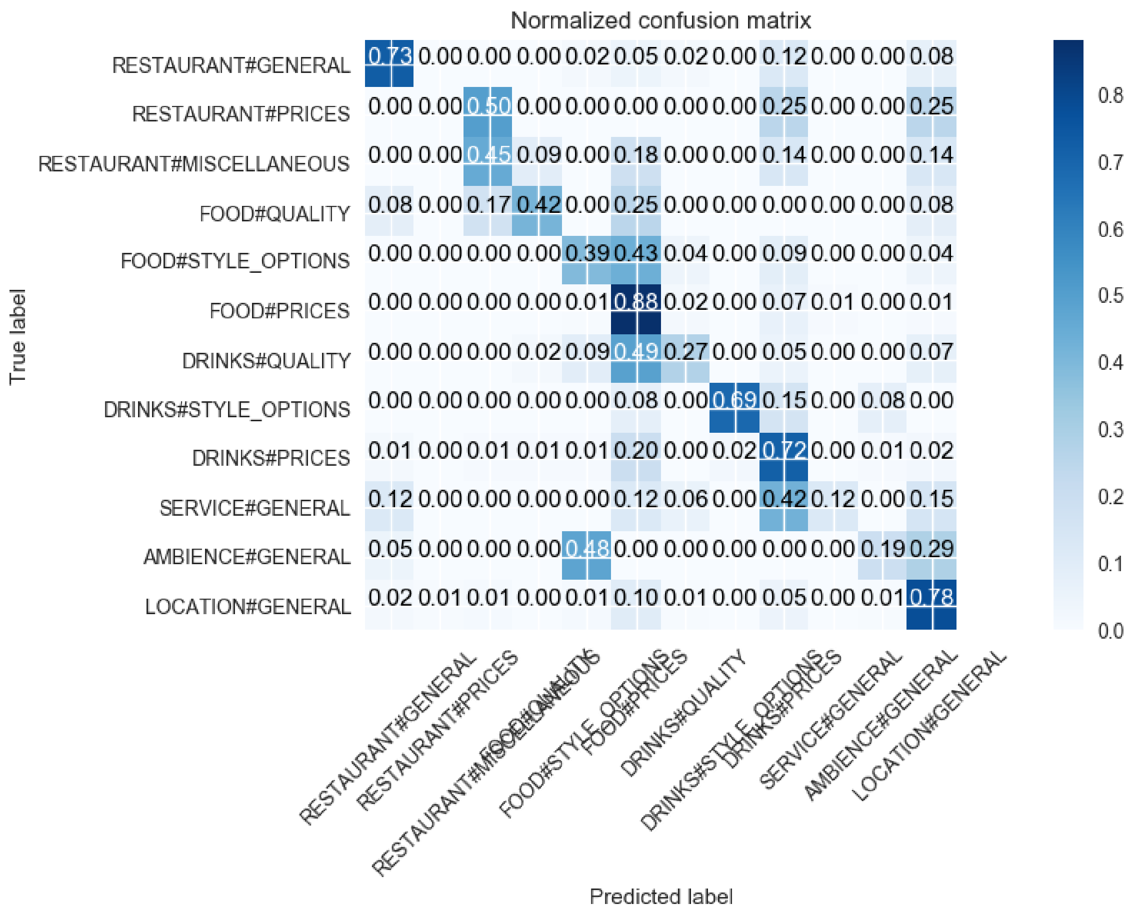

Figure 5.11: Confusion matrix for Entity#Aspect

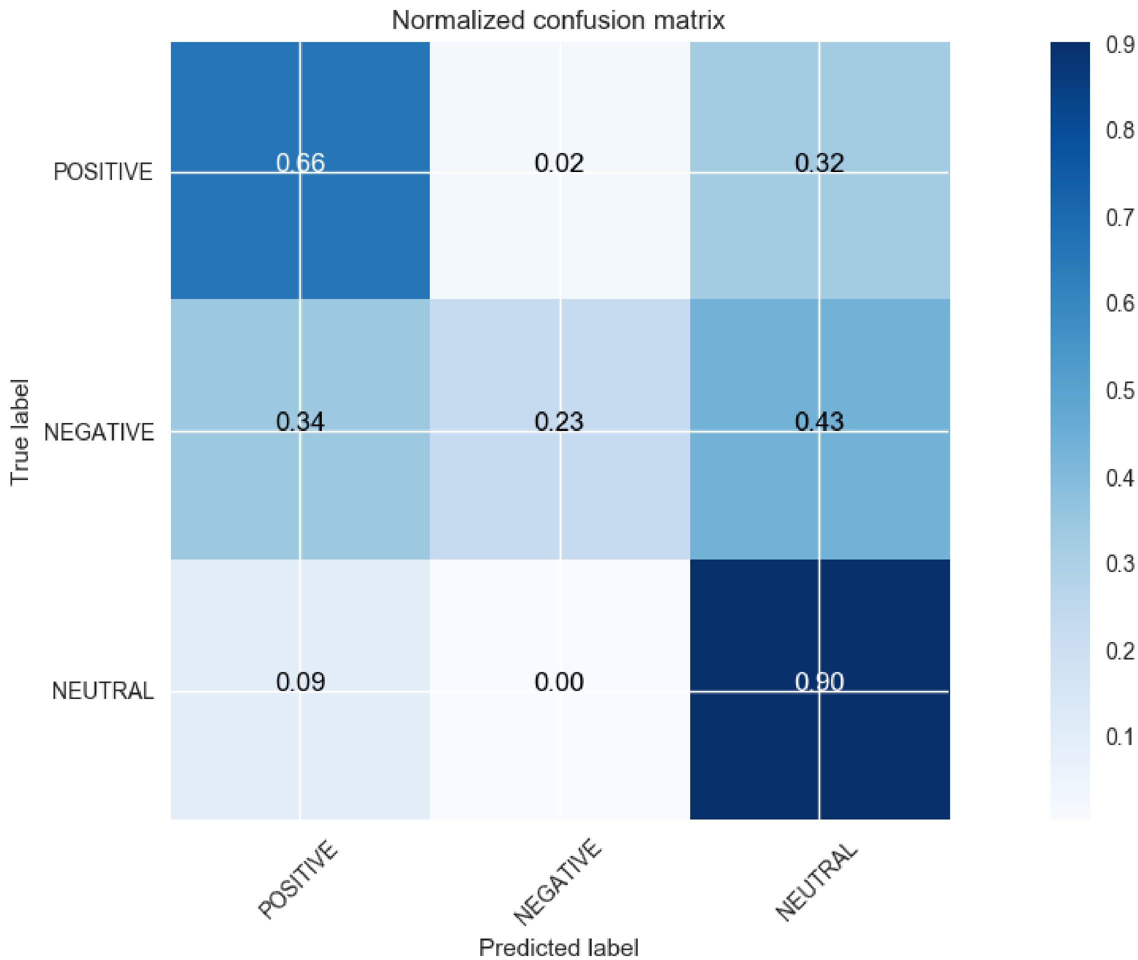

Figure 5.12: Confusion matrix for polarity clarification

5.5 Analysis

Other systems in SemEval 2016 ABSA task developed a separate binary classifier for each entities and aspects and calculated the probability for each label on the whole sentence and then used only the labels whose probability passed a threshold value. The drawback of such systems is that for a sentence with multiple entity-aspect pairs, each entity and aspect will read the signals for other entities and aspects and the probability of each entity and aspect will be less and the classifier won't be able to learn that efficiently. In the sentence "There are many vegan options, they all are delicious but a little expensive." the aforementioned systems will read the signal for "STYLE_OPTIONS", "QUALITY", and "PRICE" aspects, and each aspect will cancel the signal of every other aspect therefore reducing the probability of each aspect and thus the learning will not be accurate. But in this system in which sentences are split with an accuracy of 75%, the probability of one entity and aspect being in the split sentences is higher, which will further increase when the text simplification and splitting system will improve as discussed in the Future works section of next chapter. Thus the signal of the entity and aspect will not get canceled out by the signal of other entities and aspects, and thus the learning will be more accurate.

5.6 Conclusion

The experiments mentioned in this chapter were performed with the intention of creating a better and domain independent system for the subtasks of SemEval 2016 ABSA. From the results of the experiments, it was observed that the system was able to outperform other systems of SemEval 2016 ABSA with higher number of features when the sentences were split by this system. It might also be noted that the aforementioned system is also domain independent as it performed admirably in the restaurant domain that it was developed for while also performing competently in the laptop domain.

Chapter 6

Conclusions and future works

6.1 Conclusion

Aspect based sentiment analysis (ABSA), which is the analysis of the sentiments of every aspect of a given entity, is a field of ongoing reasearch. It was observed that most of the research have been focused on improving the efficiency of the existing ABSA systems, using different algorithms. This project was initiated with the hypothesis that a system to solve the subtasks of SemEval 2016 ABSA would perform better when the feature to split the sentences which contains multiple ENTITY#ASPECT is implemented. To test this hypothesis, a set of experiments were created that employed the feature to split the sentences along with a small number of other features. When the experiments were executed, it was observed that the system thus created was able to outperform almost all other systems for the constrained mode despite the other systems using substantially higher number of features and better classifiers. Also, the use of the feature to split the sentences had the added advantage that it made the system domain independent. It can thus be concluded from the above observations that the hypothesis that "a system to solve the subtasks of SemEval 2016 ABSA would perform better when the feature to split the sentences which contains multiple ENTITY#ASPECT is implemented" was indeed accurate.

6.2 Future works

While the performance of the system developed is acceptable, there is still space for improvement. This might be achieved by:

1. Developing an improved system to split the sentences using text simplification.
2. Implementing transfer learning by improving opinion target extraction and using it to improve ENTITY#ASPECT classification and vice-versa.

Bibliography

Nltk 3.3 documentation. http://www.nltk.org/. Accessed: 2018-04-23.

International workshop on semantic evaluation 2016. http://alt.qcri.org/semeval2016/. Accessed: 2018-03-23.

Libsvm. https://www.csie.ntu.edu.tw/~cjlin/libsvm/. Accessed: 2018-04-23.

Pystruct. https://pystruct.github.io/. Accessed: 2018-04-23.

Pytorch. https://pytorch.org/. Accessed: 2018-04-23.

Stanford parser. www.nlp.stanford.edu/software/lex-parser.shtml. Accessed: 2018-04-23.

Wordnet. www.wordnet.princeton.edu. Accessed: 2018-04-23.

T. Álvarez-López, J. Juncal-Martínez, M. F. Gavilanes, E. Costa-Montenegro, and F. J. González-Castaño. Svm and crf for aspect detection and unsupervised aspect-based sentiment analysis. *Proceedings of SemEval-2016*, (16-17):306–311, June 2016. URL https://aclweb.org/anthology/S/S16/S16-1049.pdf. Accessed: 2018-03-27.

R. Berwick. An idiot's guide to support vector machines (svms). 2003. URL http://web.mit.edu/6.034/wwwbob/svm.pdf. Accessed: 2018-04-21.

M. Buckland and F. Gey. The relationship between recall and precision. *Journal of the American Society for Information Science (1986-1998)*, 45(1):12, 01 1994. URL https://search.proquest.com/docview/216897072?accountid=13828. Copyright - Copyright Wiley Periodicals Inc. Jan 1994; Last updated - 2010-06-09.

G. D. Fabbrizio, A. Aker, and R. Gaizauskas. Summarizing online reviews using aspect rating distributions and language modeling. *IEEE Intelligent Systems*, 28(3):28–37, May 2013. ISSN 1541-1672. doi: 10.1109/MIS.2013.36. Accessed: 2018-04-31.

G. D. Fabbrizio, A. Stent, and R. Gaizauskas. A hybrid approach to multi-document summarization of opinions in reviews. In *Proceedings of the 8th International Natural Language Generation Conference (INLG)*, pages 54–63, 2014. Accessed: 2018-04-30.

S. Falk, A. Rexha, and R. Kern. Using word vectors with typed dependencies for opinion target expression extraction. *Proceedings of SemEval-2016*, (16-17):266–270, June 2016. URL https://aclweb.org/anthology/S/S16/S16-1042.pdf. Accessed: 2018-03-26.

D. Floreano. Bio-inspired artificial intelligence: theories, methods, and technologies. 2008. Accessed: 2018-04-10.

R. Gaizauskas. Lecture on sentiment analysis: Approcahes and evaluation. 2017. URL https://staffwww.dcs.shef.ac.uk/people/R.Gaizauskas/campus_only/com3110/lectures/lecture_SA3.pdf. Accessed: 2018-04-30.

T. Hofmann. Structured prediction (with application in information retrieval). 2009. URL http://mlg.eng.cam.ac.uk/mlss09/mlss_slides/Hoffman_1_2.pdf. Accessed: 2018-09-11.

D. Jurafsky and J. H. Martin. Speech and language processing. 20017. URL https://web.stanford.edu/~jurafsky/slp3/10.pdf. Accessed: 2018-04-23.

M. Kirk. *Thoughtful Machine Learning: A Test-Driven Approach*. O'Reilly Media, 2014. ISBN 9781449374105. URL https://books.google.co.uk/books?id=bSqiBAAAQBAJ.

M. Kroiss. *Predicting the Lineage Choice of Hematopoietic Stem Cells: A Novel Approach Using Deep Neural Networks*. BestMasters. Springer Fachmedien Wiesbaden, 2016. ISBN 9783658128791. URL https://books.google.co.uk/books?id=yLcqDAAAQBAJ.

F.-F. Li, J. Johnson, and S. Yeung. Convolutional neural networks for visual recognition: Convolutional neural networks. 2018. URL http://cs231n.github.io/convolutional-networks/. Accessed: 2018-04-25.

B. Liu and L. Zhang. A survey of opinion mining and sentiment analysis. pages 415–463, 2012. URL https://doi.org/10.1007/978-1-4614-3223-4_13. Accessed: 2018-03-14.

H. L. M. Dash. Feature selection for classification. *Intelligent Data Analysis*, 1:131–156, March 1997. URL https://www.sciencedirect.com/science/article/pii/S1088467X97000085. Accessed: 2018-04-06.

C. D. Manning and H. Schiitze. Foundations of statistical natural language processing. 2018. Accessed: 2018-05-01.

C. D. Manning, P. Raghavan, and H. Schiitze. Introduction to information retrieval. 2008. Accessed: 2018-05-01.

J. L. Meunier. Pystruct extension for typed crf graphs. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 04, pages 5–10, Nov 2017. doi: 10.1109/ICDAR.2017.305. Accessed: 2018-04-22.

K. P. Murphy. Machine learning: A probabilistic perspective. 2012. Accessed: 2018-04-11.

A. Ng. Cs229 lecture notes. URL http://cs229.stanford.edu/notes/cs229-notes1.pdf. Accessed: 2018-09-11.

B. Pang and L. Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2:1–135, 2008. URL http://mr.crossref.org/iPage?doi=10.1561%2F1500000011. Accessed: 2018-03-16.

M. Pontiki, D. Galanis, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, M. AL-Smadi, M. Al-Ayyoub, Y. Zhao, B. Qin, O. D. Clercq, V. Hoste, M. Apidianaki, X. Tannier, N. Loukachevitch, E. Kotelnikov, N. Bel, S. M. Jimnez-Zafra, and G. Eryiit. Aspect based sentiment analysis. *Proceedings of SemEval-2016*, (16-17):19–30, June 2016. URL http://www.aclweb.org/anthology/S16-1002. Accessed: 2018-03-25.

R. Reed and R. J. Marksll. Neural smithing: Supervised learning in feedforwarward artificial neural networks. 1999. URL https://books.google.co.uk/books?id=7C4TDgAAQBAJ. Accessed: 2018-04-20.

M. Ringsquandl and D. Petkovic. Analyzing political sentiment on twitter. In *AAAI Spring Symposium: Analyzing Microtext*, pages 40–47, 2013. Accessed: 2018-03-28.

S. Savarese. Introduction to convolutional neural networks. 2018. URL http://web.stanford.edu/class/cs231a/lectures/intro_cnn.pdf. Accessed: 2018-04-26.

R. Schapire. Machine learning algorithms for classification. URL http://www.cs.princeton.edu/~schapire/talks/picasso-minicourse.pdf. Accessed: 2018-04-11.

M. Stevenson. Lecture on information retrieval: Evaluating ir systems. 2017. URL https://staffwww.dcs.shef.ac.uk/people/R.Gaizauskas/campus_only/com3110/lectures/lecture_8_IR4.pdf. Accessed: 2018-04-30.

C. Sutton and A. McCallum. An introduction to conditional random fields. *Foundations and Trends in Machine Learning*, 4:267–373, 2011. URL http://homepages.inf.ed.ac.uk/csutton/publications/crftut-fnt.pdf. Accessed: 2018-04-21.

R. S. Sutton. Introduction: The challenge of reinforcement learning. *Sutton R.S. (eds) Reinforcement Learning. The Springer International Series in Engineering and Computer Science (Knowledge Representation, Learning and Expert Systems)*, 173:225–227, 1992. URL https://doi.org/10.1007/978-1-4615-3618-5_1. Accessed: 2018-04-09.

Z. Toh and J. Su. Nlangp at semeval-2016 task 5: Improving aspect based sentiment analysis using neural network features. (16-17):282–288, June 2016. URL http://www.aclweb.org/anthology/S16-1045. Accessed: 2018-03-26.

R. Varghese and M. Jayasree. Aspect based sentiment analysis using support vector machine classifier. In *2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 1581–1586, Aug 2013. doi: 10.1109/ICACCI.2013.6637416. URL https://doi.org/10.1109/ICACCI.2013.6637416. Accessed: 2018-03-28.

A. Vlachos. Active learning with support vector machines. pages 1–72, 2004. URL http://www.inf.ed.ac.uk/publications/thesis/online/IM040138.pdf. Accessed: 2018-04-18.

A. Vlachos. An investigation of imitation learning algorithms for structured prediction. In M. P. Deisenroth, C. Szepesvri, and J. Peters, editors, *Proceedings of the Tenth European Workshop on Reinforcement Learning*, volume 24 of *Proceedings of Machine Learning Research*, pages 143–154, Edinburgh, Scotland, 30 Jun–01 Jul 2013. PMLR. URL http://proceedings.mlr.press/v24/vlachos12a.html. Accessed: 2018-04-15.

A. Vlachos. Lecture on language modeling basics. 2018. URL https://sheffieldnlp.github.io/com4513-6513/lectures_reveal_js/lecture2_basicLMs.html. Accessed: 2018-04-28.

L. Xiang-wei and Q. Yian-fang. A data preprocessing algorithm for classification model based on rough sets. *Physics Procedia*, 25:2025 – 2029, 2012. ISSN 1875-3892. doi: https://doi.org/10.1016/j.phpro.2012.03.345. URL http://www.sciencedirect.com/science/article/pii/S1875389212007614. International Conference on Solid State Devices and Materials Science, April 1-2, 2012, Macao.

W. Zhaowen, Y. Jianchao, and Z. Haichao. *Sparse Coding And Its Applications In Computer Vision*. World Scientific Publishing Company, 2015. ISBN 9789814725064. URL https://books.google.co.uk/books?id=cmLFCwAAQBAJ.