Implicit aspect-based opinion mining and analysis of airline industry based on user generated reviews

Vaibhav Oberoi
School of Computing
Dublin City University
Dublin, Ireland
vaibhav.oberoi2@mail.dcu.ie
19210149

Abstract – Mining opinions from reviews has been a field of ever-growing research. These include mining opinions on document level[1], sentence-level, and even aspect level[2] of a review. While explicitly mentioned aspects in a review have been widely researched, very little work has been done in gathering opinions on aspects that are implied and not explicitly mentioned. E.g. *"the flight was spacious and there was plenty of legroom"*. This gives an opinion on the entities of the *cabin* and *seat* of an airline. Words like "*spacious*" and phrases like "*plenty of legroom*" help identify these implied entities and the opinions attached to them. Not much research has been done for gathering such implicit aspects and opinions for airline reviews. The present study aims to extract and analyse opinions about such implied aspects and entities of airlines. As part of this effort, we were able to develop a domain-specific aspect-based corpus[3]. Also, a novel method using stochastic gradient descent[4] with L2 regularization[5] of *conditional random field* coupled with *machine learning* and *ensemble learning* classification techniques to identify and extract such implied aspects and entities.

Index Terms – Reviews, Opinions, Implicit aspects, Airline, Conditional Random Field, Machine Learning, Ensemble learning, Classification, Identification, Extraction, Corpus

I. Introduction

Travel and tourism are well-liked terms amongst all generations of people. The airline industry is a key facilitator in this domain. For this industry, serving its customers with not only cost-effective but also satisfactory service options is paramount. Gone are the days when passengers were required to fill feedback forms during their journey. In this 21st information age, with constant development in social and web media, a multitude of platforms are available like *Trip Advisor*, *Airline Ratings*, etc, for consumers to express their views on air travel. This also serves in favour of the airline companies, as it becomes their one-stop to access rich customer feedback information. Also, opinions are very important to businesses and organizations because they always want to find consumer or public opinions about their products and features. [1]

Many times, due to a variety of reasons like paid promotions, fraudulent, and even unstructured nature of these reviews, insightful information cannot be extracted. So, a need is felt to have a mechanism that can gather cognizance in terms of the perception of customers on airline-specific aspects. The present study provides a mechanism to gather opinions and aspects from such reviews which are not explicitly mentioned

Liu and Zhang et. al. defined the term opinion as a concept covering sentiment, evaluation, appraisal, or attitude held by a person. [1] Aspects and entities are more like topics in a text document. *Hu and Liu et. al*. coined this type of analysis as feature-based sentiment analysis. [2] Aspect or entity-based analysis identifies the

[1] *Document level: A whole document consisting of multiple paragraphs and sentences, Example a book.*
[2] *Aspect level: In simple terms, it means as a topic in a sentence*
[3] *Corpus: Collections of texts formed by a single language*
[4] *Stochastic gradient descent is an iterative method for mathematical optimization to select best elements from a set*
[5] *L2 regularization is a nomenclature for ridge regression, which adds a penalty term which is sum of squares of all feature weights.*

target of the opinion. It is a fine-grained approach for text analysis.

Regarding the present study, an entity is the feature of the airline. Example, *food, cabin, seat, staff*, etc. Since these entities in themselves can have various attributes associated with them, it becomes important to divide them further into sub-aspects. For example, *the cabin* is not always referred independently, it has its attributes like *space, condition*, and *temperature* mentioned along with it. For example, a review for cabin looks like "*the cabin was cold and a bit weary*". Here opinion terms like "*cold"* and "*weary"* become sub-aspects for temperature and condition respectively. This approach is used in this present study to make a fine-grained analysis of opinions and map them accurately to the respective entity sub-aspect pair.

The present study addresses following key research questions.

R.Q.1. *Which airline industry specific aspects can be leveraged for implicit aspect-based opinion mining?*
R.Q.2. *How can an annotated domain-specific implicit aspect-based sentiment corpus be engineered that targets the reviews of Airlines?*
R.Q.3. *What is the specific sentiment lexicon*[6] *generation techniques that could influence implicit aspect-based opinion mining tasks?*
R.Q.4. How can performance be improved for implicit aspect-based extractions and opinion classification?

Trip Advisor is an online microblogging platform primarily used for viewing reviews and experiences of travellers either travelling to the same destination or other all over the globe. *Airline Ratings* is another website that boasts of maintaining a repository of reviews of airlines. Usually, people before making airline ticket purchases do read reviews.

In the present study, using the *selenium framework*, a python-based bot was developed to scrape these reviews meticulously without extracting any personal information of review authors to comply and adhere to GDPR laws.

In this study, 3000 reviews were collected within a time period of 1 month (from 1st November to 30th November 2019) with an aim to study public's opinion with respect to 16 Airlines.

In summary, the goal of this study is to extract implied aspects and opinions from reviews. This can be achieved through an approach which involves using a few state-of-the-art modules like *Stanford's Core NLP Parser*, *NLTK's Vader Sentiment Analyser*, *Sci-Kit Learn Python Libraries.*

II. Related Work

The present study concentrates on implicit aspect extraction, opinion lexicon generation, and engineering an annotated implicit aspect-based sentiment corpus that can influence implicit opinion mining from consumer reviews in the airline industry. Few studies that are done in this realm of implicit aspect-based opinion mining and extraction but very few on implicit aspect-based opinion mining.

In a research study proposed by *Chinsha T C et al.* the methodology proposes a syntactic based approach using dependency parsing[7].[3] In another research for comparing word representations for implicit classification.[4] Both these studies use *SentiWord Net* and have dataset restrictions. The present study intends to extend the results of these two papers. By using a syntactic approach to group implicit aspect synonyms for a larger dataset. As the two studies were restricted to 170 and SemEval dataset respectively.

Research dealing with the double-implicit problem[8] in opinion mining and sentiment analysis proposes a protocol to derive a labelled corpus for implicit polarity and aspect analysis. [5] The work in this paper is limited to only Chinese restaurant reviews. The present study addresses not only the dataset limitation but also the labelling of corpus technique by using Type Token Ratio and other corpus statistic techniques which are explained in the experimental setup section.

Another study using two corpora proposed a hybrid model to support Naïve Bayes training to identify implicit aspects. [6] This corpus and dictionary-based approach is limited to only

[6] Lexicon: It is a component of natural language processing that contains grammatical information about individual words or strings

[7] Dependency parsing: A methodology that is used to extract grammatical structure from sentences.

[8] Double-implicit: Word or word-phrases that not only describe an entity but also the opinion of the entity.

adjective type words of a sentence. The present study extends this work by taking considering a combination of adjectives, adverbs, nouns, and other part-of-speech indicators and uses ensemble learning for classification.

A study conducted on implicit aspect indicator extraction, models relations between the polarity of a document and its opinion target using Conditional Random Field. [7] The Conditional Random Field method proposed in this study is limited to only cellular device data and the entities are picked from a pre-trained Stanford CRF model. The present study will use specifics from the methodology of introducing Conditional Random Field and extend it to the airline domain.

III. Methodology

Since the present study uses a supervised learning method and the data being one of a kind, needs to be labelled. An inter-annotator agreement and annotation guidelines between the annotators is setup.[8] (Appendix A)

The objective of annotation is to label words at two levels i.e. entity-level and implied aspect-level labelling. A detailed explanation is available in the experimental setup section. The level of agreement between annotators is evaluated by using Cohen's Kappa coefficient.[9] Details of which can be found in the evaluation section.

The methodology is divided into two methods, one to identify and extract entities by a sequence labelling task using Conditional Random Field, and the other is to identify and classify implicit aspect groups.

So, the feature engineering task is also divided into two modules. One is to engineer and extract word features and the other is to engineer numeric features for word representations. Word features are generated from text reviews by making use of syntactic features like parts-of-speech and dependency parsing. (Appendix B)

Also, experimental setup section provides examples of the word feature generation.

Numeric representation of words is done by computing three techniques, viz, count vectorization, term frequency-inverse frequency, and word embeddings. (Appendix C). The implementation details for these methods is discussed in the experimental setup section.

The numeric representation techniques are frequency-based and lack semantic and contextual representation of the words. This can be achieved by word embeddings. [10] But, in this experimental study, the dataset is limited to only a few thousand reviews, so if a word embeddings model is trained only on these reviews, the results will not be effective. So, pre-trained glove vectors [11] that are trained on user-generated data is used.

This experiment study takes this contextual representation step further and creates a custom word embedding model which not only has Glove embeddings but also word embedding vectors for this study's corpus. The detailed implementation can be found in the experimental setup section. (Appendix D)

Sequence Labelling using Conditional Random Fields

Sequence Labelling: It is a supervised learning[9] task where label is assigned to each element of a sequence.

Let $X = (x_1, x_2, x_3, \dots, x_n)$ be a set of n observations and $Y = (y_1, y_2, y_3, \dots, y_n)$ be a set of assigned labels to these n observations. The motive is to be able to predict a set of labels $Y^T = (y_1, y_2, y_3, \dots, y_n)$ given a new set of inputs $X^T = (x_1, x_2, x_3, \dots, x_n)$ with a supervised machine learning model trained on these inputs X and labels Y. Conditional Random Field has been identified the machine learning methodology for this experiment study.

Before digging in deep in working of conditional random fields and specifically for this experiment study, there are few terminologies like generative and discriminative probabilistic models, Hidden Markov models (HMMs) and sequential classification that need to be understood. (Appendix E)

CRFs can adjust to a variety of statistically correlated features as input just like a sequential classifier. And just like a generative probabilistic model it trades-off decisions at different sequence in order to obtain a global optimal labelling.

[9] Supervised learning: It means learning a mapping between a set of input variables X and output variables Y and applying these mappings, predictions can be made for unseen data

Definition of CRF:

Let $G = (V, E)$ denote a graph such that $Y = (Y_v)_{v \in V}$, so that Y is indexed by the vertices of G. Then (X, Y) is a conditional random field in case, when conditioned on X. the random variables Y_v obey the Markov Property with respect to the graph. [12] The equation can be denoted as

$$p\,(Y_v|X, Y_w, w \neq v) = p\,(Y_v \mid X, Y_w, w \sim v) \text{ - eq1}$$

where, $w \sim v$ means that w and v are neighbours in G and w and v belong two different sequences of data.

The present study uses a case of Stochastic Gradient Descent (SGD) with L2 regularization. [13]

The way this works is as follows,

Being a linear model, the training objective function[10] has the form, [14]

$$E_n(w) = \frac{\lambda}{2}\,|w|^2 + \frac{1}{n}\sum_{i=1}^{n} l(y_t.w.x_t)\,[14] - \text{eq2}$$

Where $y_t = \pm 1$ and where the function $l(m)$ is convex. The corresponding stochastic gradient update is then obtained by approximating derivative of the sum by the derivative of the loss with respect to a single example, as follows [14]

$$w_t = (1 - \gamma_t \lambda)w_t - \gamma_t y_t x_t l^2(y_t x_t w_t)[14] - \text{eq3}$$

where, γ_t is the learning rate l^2 is the L2 penalty term.

In simpler terms to maximize the likelihood of CRF, it can be represented as

$$\log P(Y|X) = w\,.\,\varphi(Y, X) - log \sum_{Y^T} e^{w\varphi(Y,X).}$$

$$= w\,.\,\varphi(Y, X) - logZ$$

Taking derivatives,

$$\frac{d}{dw}\log P(Y|X) = \varphi(Y, X) - \frac{d}{dw}\log \sum_{Y^T} e^{w\varphi(Y,X).}$$

$$= \varphi(Y, X) - \frac{1}{Z}\sum_{Y^T} \frac{d}{dw} e^{w\varphi(Y,X).}$$

$$= \varphi(Y, X) - \sum_{Y^T} \frac{e^{w\varphi(Y,X).}}{Z}.\varphi(Y, X)$$

$$= \varphi(Y, X) - \sum_{Y^T} P(Y^T|X)\varphi(Y^T, X)$$

This is

$$\frac{d}{dw}\log P(Y|X) = \varphi(Y, X) - \sum_{Y^T} P(Y^T|X)\varphi(Y^T, X)$$

Where it means $\varphi(Y, X)$ to add correct feature and subtract $P(Y^T|X)$ which is expectation of features.

And adding a l2 term penalty makes this L2 Stochastic gradient descent.

After processing training reviews on this novel approach of using conditional random field with stochastic gradient descent with L2 regularization the entities are mapped to the sequence of training data.

Once this is mapped, the methodology moves to the next step of classification.

Classification Algorithms

The aspect extraction task needed classifier models that could accurately predict the aspect. Different algorithms were used to classify and compare how accurate each model was to classify these sub-aspects. Algorithms like Support Vector Machine, Decision Trees, Random Forest, a bagging ensemble learning algorithm Voting Classifier and a boosting ensemble learning algorithm XGBOOST were used.

Support Vector Machine

It is a dependable and fast classification algorithm that visualizes the data as points on a space characterized by its features. It uses a hyperplane to segregate and categorize the data as per the classes.[15] (Appendix F)

Decision Tree

It is a recursive classification procedure that partitions dataset into smaller groups based on a set of tests defined at each branch. [16] (Appendix F)

[10] Objective function: Refers to a mathematical function to be maximized or minimized in an optimization problem.

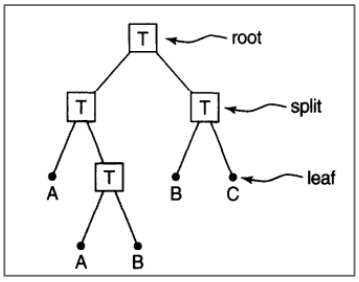

Fig.1. A decision tree classifier.

Random Forest

It is essentially an ensemble classifier that uses several decision trees and outputs the class that is predicted by the maximum number of trees. It is not dependent on any decision tree, instead it is dependent on a bunch of them making it robust. It is a way to decrease variance of the prediction by generating supplementary data from training set using several combinations with repetitions. This implements Brieman's bagging technique. [17]

Voting Classifier

It is an ensemble learning method. It is a wrapper of a set of different algorithms which are trained and evaluated parallelly to make use of features of each algorithm.[18]

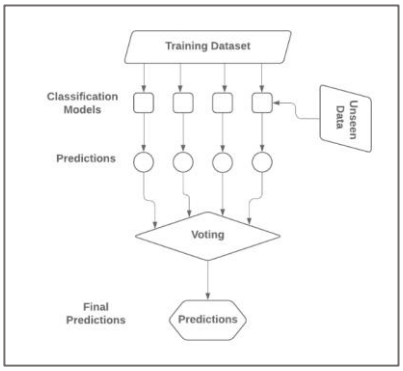

Fig. 2. Voting Classifier

In the present study, three classification algorithms are wrapped within the voting classifier. They include decision trees, random forest and Extra Trees Classifier.

XGBOOST

Boosting means training a sequence of classifiers one after another so that the last classifier is trained in a better and efficient manner to predict class labels for examples the previous classifiers performed poorly on.

Algorithm 1 Boosting Algorithm

Input *Full training set of N examples; maximum ensemble size T; sample size L ¡ N.*

Approach to Boosting

Assign an equal weight of 1/N to all training examples

for i=1 to T **do**

a) based on current weights, randomly sample L examples from training set without replacements

b)train classifier on this sample

c) identify misclassifed examples

d) increase weights for misclassified examples

Output: ***Final Model based on all classifiers***

Algorithm 1. Boosting Algorithm

Using this approach, the present study identified XGBOOST as the boosting algorithm to classify sub-aspects.[19] (Appendix F)

IV. Experimental Setup and Discussions

Data Scraping: After getting permission from Trip-Advisor and Airline Ratings website, making use of Selenium web-drivers, bots were created.[20] Any form of personal data or personal identity identifying data such as review author's name, review author's id, timestamp of review, review number and even flight ticket details was not extracted. This was done to adhere and abide by the GDPR laws.

Data Pre-processing: Using standard pre-processing techniques like removing domain-specific stop words, removal of unnecessary punctuations, spell correction, converting numbers to words, and word standardization. The motivation for doing so was to avoid misleading the training model. Also, since the data was user generated, there were many contractions of words, for example, couldn't, can't, aren't, I'm etc were seen quite often in the texts. So, fixing these contraction words was also a part of the study. Words like couldn't were replaced by could not. (Appendix G)

Corpus Statistics: The data being user generated was raw and unstructured. It is the first time this group of reviews were considered text mining and analysis. So, two statistical strategies, viz, type token ratio and Zipf's distribution were used to determine variability in dataset.

Type Token ratio (TTR) is represented as follows, (Appendix H)

$$TTR = \frac{(number\ of\ types)}{(number\ of\ tokens)}$$ [23]

Table 1

Type Token Ratio Scores

Source	TTR Score

Trip Advisor	0.35
Airline Ratings	0.37

TTR Scores are low for both data sources, this means that there are many repeated terms in the corpus.

More detailed TTR scores can be found in the appendix.

Zipf's law states that a relationship between frequency of word (f) and its position in the list i.e. its rank (r) is inversely proportional to one another. [21] (Appendix H)

$$f \propto \frac{1}{r}$$

Manual Annotation: As explained in the methodology, annotation was done on two levels using Doccano software[22]. There are detailed examples and explanation of this manual annotation strategy.

Table 2
Detailed example of Level 1 annotation

INPUT: "Overall the experience was comfortable and spacious with delicious meals"

Output: [("experience was comfortable", "Inflight"), ("spacious", cabin), ("delicious meals", "food)]

Once, entity-level tuples[11] were tagged containing word or word phrases with entity-name, as seen in above example. After completing entity level annotation, another fine-grained approach to classify entity-wise word or word phrases to their respective implied aspects was conducted.

Table 3
Detailed example of Level 2 annotation

INPUT: [("experience was comfortable", Inflight), ("spacious", cabin), ("delicious meals", "food)]

OUTPUT: [[("experience", inflight-operations), ("comfortable", inflight-operations)], [("spacious", cabin-size)], [("delicious", food-taste), ("meals", food-service)]]

Cohen's Kappa Co-efficient and Inter Annotator Agreement: As explained in methodology of this experiment study, after adhering with the guidelines in the inter-annotator agreement, and using Sklearn Kappa score library, the Cohen's Kappa score for level of agreement was calculated.

Training Data Preparation: The experiment study used techniques described in the methodology section for preparing the training data. Taking an example sentence, this process will be explained in detail.

Example sentence: "*Overall, the experience was comfortable and spacious with delicious meals."*

Table 4
Annotated and labelled list of example sentence

Entity Level	
Entity	Word/Phrases
In-flight service	Experience was comfortable
Cabin	Spacious
Food	Delicious meals
Implicit Aspect Level	
Aspect	Word
Inflight Operations	Experience
Inflight Operations	Comfortable
Cabin Size	Spacious
Food Taste	Delicious
Food Service	Meals

From this review, words like *experience*, *comfortable*, *spacious*, *delicious*, and *meals* were identified as aspect terms and their semantic and syntactic information was extracted by parsing them through off-the shelf state of the art models like Stanford Core NLP to extract part-of-speech tags and dependency tags, and Vader for sentiment score

Part-of-speech and Dependency Tags:

Table 5
POS-tags using Stanford Core NLP

Input: "*Overall the experience was comfortable and spacious with delicious meals"*
Pre-processing: "*overall experience comfortable spacious delicious meals"*
POS-Tags: [('*overall*', 'JJ'), ('*experience*', 'NN'), ('*comfortable*', 'JJ'), ('*spacious*', 'JJ'), ('*delicious*' , 'JJ'), ('*meals*', 'NNS')]

Here the tags "JJ", "NN" and "NNS" mean adjective, noun and singular noun respectively. (Appendix B)

[11] Tuples are a data type that is similar but also distinct to the list data type. The instances are characterized by having fixed attributes and the elements of a tuple instance can differ in data type amongst one another.

The dependency graph tree for above pre-processed sentence can be visualised using GraphViz as below,

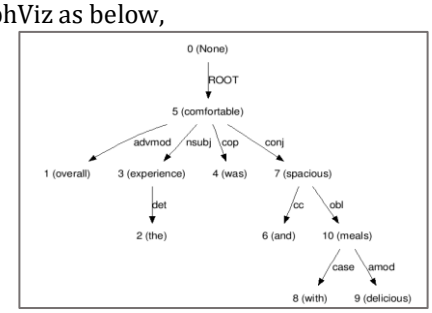

Fig 3. Dependency Tree Graph

The information from this structure is captured in a list of tuples and stored in a data frame in python as below, (Appendix B)

Vader Sentiment Score: Being a simple rule-based model which generates sentiment analysis for social media texts, sentiment score for label words and their dependent words was added to the tuple. [23]

Using these techniques, following information for labelled words was extracted in the form, (Main-word, Main-word POS Tag, Dependent word, Dependent word POS Tag, Main-word sentiment score, Dependent Word sentiment score, Dependency Tag, Previous, Next Word)

List of tuple features: [("experience", "NN", "comfortable", "JJ", 0.0, 0.4, "amod", "overall", "comfortable"), ..., ("delicious", "JJ", "meals", "NNS", 0.6, 0.0, "advmod", "spacious", "meals)]

For the task of sequence labelling in order to identify the entity a word or word phrase belongs to, the tuples were added with their respective labels i.e. the label added to a tuple was the label that word belonged to.

For example, Tuple, ("delicious", "JJ", "meals", "NNS", 0.6, 0.0, "advmod", "spacious", "meals) has main word *food*, so a new entry to this was made as *"f"*, which became the *Y* or dependent variable.

After getting results from the CRF model, the entity-id i.e. if it was classified as "*food*" then *id* was "*f*".

Table 6
Entity ID List

Entity	Entity-ID
Food	f
Cabin	c
Entertainment	e
Staff	st
Seat	s
Off-flight	o
In-flight	i
Possession	p

Once the correct entity is identified, the next step is to classify which aspect is mentioned in the sentence. Then to the word feature tuple ENTITY-ID is added to the training data and it is then vectorized.

Vectorization: The main word, dependent word, previous word and next word are replaced by their numeral values.

Count Vectorization: For this experiment study, since the methodology does try to keep certain punctuations and special characters, a need is felt to create own tokenizer. The results for an example sentence
Sentence: *"so overall I highly recommend this airline"*
Vectors: {"*so*":5, *"overall"*:3, *"I"*:2, *"highly"*:1, *"recommend"*:4, *"this"* :6, *"airline"* :0}

TF-IDF Vectorization: For this experiment study, TF-IDF score for the words in the feature sets was calculated using sci-kit learn tf-idf vectorizer. Below, is the result of tf-idf scores for all corpus words.

Table 7
TF-IDF Vectorization

Word	TF-IDF score
Basic	0.965545
Redemption	0.965545
Haul	0.962570
Rescue	0.958253
receipt	0.948869
Lovely	0.943597
Hindsight	0.929212
Prior	0.928020

Word Embeddings: As mentioned in the methodology, the corpus of this experiment study was small. So, a word embedding model using Word2Vec for the corpus was trained. And a pre-trained Twitter Glove Embeddings consisting of a vocabulary size of 1.2 million words and 27 billion tokenized twitter words with a 100-dimensional vector was selected.
Using the below algorithm, a set of new vector embeddings were merged using pre-trained Glove and corpus Word2Vec embeddings.

Algorithm 1 *Word embedding vector generation using pre-trained glove vectors*

Inputs

$S = [W_1, W_2, W_3, ..., W_n]$, *Input sentence S contains n words*

path = path where downloaded embeddings are stored

GloveVec = Pretrained Glove Vectors

Output

Word2Vec Embedding Model

```
// Load the Glove Pre-trained Vectors
with open(path):
  gloveVec = embedding vectors // Create Word2Vec Embedding for Airline Corpus
word2vec = Word2Vec Create Model

for word, vector in zip(word2vec.index2word, word2vec.vectors) do
 | w2v = dict(word: vector)
end
// Vectors for airline Corpus are loaded
for each Wi in S do
 | if Wi exists in gloveVec then
 |  | extract vecWi  MVi = vecWi end                                  end
 | else if Wi exists in w2v then
 |  | extract vecWi  MVi = vecWi end
 | else
 |  | extract vecWi  MVi = generateNewvecWi end
```

Algorithm 2. Custom word embedding algorithm

With this algorithm, a new set of word embeddings were generated to vectorize main, dependent, previous and next words.

Cosine Similarity Index: Along with the word embeddings, cosine similarity between main and dependent word was added as a new feature. (Appendix D)

These new features were then used to classify opinionated texts into their respective implicit-aspect classes.

Handling Class Imbalance: After annotation, there was high imbalance amongst implicit aspect classes of almost all entities. The imbalance for entity cabin, can be seen below.

Class: {"*Condition"*:182, *"Size"* :61, *"Temperature"* :39, *"Fragrance"* :20}

This imbalance was handled using an oversampling technique called SMOTE.

Results of SMOTE imbalance handling is as follows,

Class: {"*Condition"*:182, *"Size"*:182, *"Temperature"*: 117, *"Fragrance"*: 102}

This could be visualized as a scatter distribution shown below

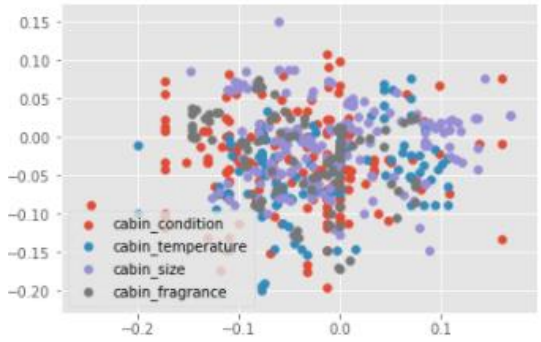

Fig Scattered class distribution after handling imbalance using SMOTE

Implicit Aspect Classification: A total of 8 models were created for each entity i.e. there are independent classification models for training each entity.

This experiment study makes use of state-of-the-art classification algorithms. Three of which were ensemble learning techniques. These include, Gradient boosting algorithm – XGBOOST, a Voting Bagging algorithm using three trees based classification techniques like Decision Trees, Random Forest and Extra Trees Classifier. And other machine learning techniques like SVM, Decision Tree.

The reason for using these different algorithms was to gather insightful information on the performance of classification which was evaluated based on ROC-AUC score and F1 scores. (Appendix I)

V. Evaluation and Results

This experiment study using state-of-the-art techniques and algorithms is a new approach to mine and extract implicit aspects from opinionated texts.

First evaluation was for the annotation of the dataset using Cohen's Kappa Co-efficient.

The two annotators agreement score ranged from 0.77 to 0.80 for entity level and implicit aspect level annotation

Second evaluation was for the sequence labelling task using a stochastic gradient descent with L2 regularization Conditional Random Field. This was done to classify texts in 8 different entities.

The ROC-AUC score achieved for this task is **96.5%** and a F1 score of **94.56%**. (Appendix I)

Third evaluation was for the classification task using five different classification algorithms.

A detailed ROC-AUC score evaluation metric is available below

Table 8
ROC-AUC Scores for classification of entities

Entity	Algorithms				
	S	D	R	V	X
Food	84%	92%	94%	94.8 %	94.7 %
Cabin	75 %	75 %	85%	85.6 %	77 %
Entertainment	73.6 %	79.9 %	83.1 %	84.3 %	85.9 %
In-flight	60.3 %	70.3 %	72.2 %	74.9 %	71.2 %
Off-flight	66.4 %	86.2 %	84.9 %	84.8 %	89.8 %
Possession	66.9 %	66.9 %	70.5 %	73.3 %	73.4 %
Seat	66 %	73.7 %	75 %	75.7 %	78%
Staff	75.6 %	76.9 %	80.9 %	82.1 %	81.4 %

In the above table, S stands for Support Vector Machines, D for Decision Trees, R for Random Forest, V for Voting Classifier and X for XGBoost algorithms

In all these machine learning and ensemble learning classification algorithms, the bagging technique of ensemble using tree-based classifiers has out-performed all other classification algorithms. (Appendix I)

VI. Conclusion and future work

The present research study using a supervised machine learning approach answers all the research questions posed at the start successfully.

R.Q.1. Answer: Eight different airline industry specific aspects can be leveraged. They include fine-grained entities like cabin, entertainment, food, in-flight service, off-flight service, seat, staff and possessions.

R.Q.2. Answer: Manual annotation with help of open source tool like Doccano resolves the annotation hurdle. The annotation is done on two levels, one is on the entity level and the other is on the sub-aspect level. The two annotators in this experiment study have a very good agreement on annotated terms. This can be reflected by a Cohen's Kappa score ranging from 0.77 to 0.80. So, it can be said that the corpus derived from this study, can be used as a gold standard for implicit aspect-based mining tasks for airline reviews.

R.Q.3. Answer: This experiment study presents a novel approach of dividing the implicit aspect-based opinion mining task in two levels, one using stochastic gradient descent with L2 regularization for improving conditional random fields to identify entities. This is done with a ROC-AUC Score of 96.58%, a F statistic score of 94.56% and with 0.01 degrees of mean absolute error on testing data. The second level is to classify each entity into implicit aspect sub-groups. For this state-of-the-art machine and ensemble learning algorithms are used. In this experiment study, it is found that ensemble learning algorithms both bagging and gradient boosting have outperformed the machine learning algorithms. The ROC-AUC scores for ensemble learning algorithm like Voting Classifier ranges from 73% to 94.8% and the boosting algorithm like XGBOOST ranges from 71% to 94.7&

R.Q.4. Answer: Since the experiment study was constrained to a limited size of user generated reviews, there was a very high imbalance in the training dataset. This was a performance challenge encountered during the study. But by using Synthetic Minority Oversampling Technique, this challenge was overcome.

The scope of this experiment study is limited to a few reviews, as a possible future work, another study can carry forward the methods proposed in this paper to a larger dataset. Also, another possible future work can be implementing a neural architecture of these proposed methods.

VI. References

[1] B. Liu and L. Zhang, "A Survey of Opinion Mining and Sentiment Analysis," in *Mining Text Data*, C. C. Aggarwal and C. Zhai, Eds. Boston, MA: Springer US, 2012, pp. 415–463.

[2] M. Hu and B. Liu, "Mining and Summarizing Customer Reviews," p. 10.

[3] C. T. C and S. Joseph, "A syntactic approach for aspect based opinion mining," in *Proceedings of the 2015 IEEE 9th International Conference on Semantic Computing (IEEE ICSC 2015)*, Feb. 2015, pp. 24–31, doi: 10.1109/ICOSC.2015.7050774.

[4] C. Braud and P. Denis, "Comparing Word Representations for Implicit Discourse

Relation Classification," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, Sep. 2015, pp. 2201–2211, doi: 10.18653/v1/D15-1262.
[5] H.-Y. Chen and H.-H. Chen, "Implicit Polarity and Implicit Aspect Recognition in Opinion Mining," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Berlin, Germany, Aug. 2016, pp. 20–25, doi: 10.18653/v1/P16-2004.
[6] E. H. Hajar and B. Mohammed, "Hybrid approach to extract adjectives for implicit aspect identification in opinion mining," in *2016 11th International Conference on Intelligent Systems: Theories and Applications (SITA)*, Oct. 2016, pp. 1–5, doi: 10.1109/SITA.2016.7772284.
[7] I. Cruz, A. Gelbukh, and G. Sidorov, "Implicit Aspect Indicator Extraction for Aspect-based Opinion Mining," p. 18.
[8] "Natural Language Annotation for Machine Learning [Book]." https://www.oreilly.com/library/view/natural-language-annotation/9781449332693/ (accessed Aug. 16, 2020).
[9] A. Rosenberg and E. Binkowski, "Augmenting the kappa statistic to determine interannotator reliability for multiply labeled data points," in *Proceedings of HLT-NAACL 2004: Short Papers*, Boston, Massachusetts, USA, May 2004, pp. 77–80, Accessed: Aug. 15, 2020. [Online]. Available: https://www.aclweb.org/anthology/N04-4020.
[10] Y. Goldberg and O. Levy, "word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method," *arXiv:1402.3722 [cs, stat]*, Feb. 2014, Accessed: Apr. 22, 2020. [Online]. Available: http://arxiv.org/abs/1402.3722.
[11] J. Pennington, R. Socher, and C. Manning, "GloVe: Global Vectors for Word Representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, Oct. 2014, pp. 1532–1543, doi: 10.3115/v1/D14-1162.
[12] J. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," p. 10.
[13] N. Sokolovska, T. Lavergne, O. Cappé, and F. Yvon, "Efficient Learning of Sparse Conditional Random Fields for Supervised Sequence Labelling," *IEEE J. Sel. Top. Signal Process.*, vol. 4, no. 6, pp. 953–964, Dec. 2010, doi: 10.1109/JSTSP.2010.2076150.
[14] L. Bottou, "Stochastic Gradient Descent Tricks," in *Neural Networks: Tricks of the Trade*, vol. 7700, G. Montavon, G. B. Orr, and K.-R. Müller, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 421–436.
[15] J. A. K. Suykens and J. Vandewalle, "Least Squares Support Vector Machine Classifiers," *Neural Processing Letters*, vol. 9, no. 3, pp. 293–300, Jun. 1999, doi: 10.1023/A:1018628609742.
[16] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 21, no. 3, pp. 660–674, May 1991, doi: 10.1109/21.97458.
[17] A. Cutler, D. R. Cutler, and J. R. Stevens, "Random Forests," in *Ensemble Machine Learning: Methods and Applications*, C. Zhang and Y. Ma, Eds. Boston, MA: Springer US, 2012, pp. 157–175.
[18] S. Saha and A. Ekbal, "Combining multiple classifiers using vote based classifier ensemble technique for named entity recognition," *Data & Knowledge Engineering*, vol. 85, pp. 15–39, May 2013, doi: 10.1016/j.datak.2012.06.003.
[19] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco California USA, Aug. 2016, pp. 785–794, doi: 10.1145/2939672.2939785.
[20] S. Munzert, C. Rubba, P. Meißner, and D. Nyhuis, *Automated Data Collection with R: A Practical Guide to Web Scraping and Text Mining*. John Wiley & Sons, 2014.
[21] D. M. W. Powers, "Applications and Explanations of Zipf's Law," 1998, Accessed: Aug. 16, 2020. [Online]. Available: https://www.aclweb.org/anthology/W98-1218.
[22] *doccano/doccano*. doccano, 2020.
[23] C. J. Hutto and E. Gilbert, "VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text," May 2014, doi: null.