

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/365352362>

Multi-aspect Sentiment Analysis Using Domain Ontologies

Chapter · November 2022

DOI: 10.1007/978-3-031-21422-6_19

CITATIONS

0

READS

21

3 authors:



Srishti Sharma
Amity University

18 PUBLICATIONS 29 CITATIONS

[SEE PROFILE](#)



Mala Saraswat
ABES Engineering College

27 PUBLICATIONS 84 CITATIONS

[SEE PROFILE](#)



Dr Anil Kumar Dubey
ABES Engineering College

65 PUBLICATIONS 113 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Emotion based trust networks, recommender systems and NLP [View project](#)



2nd International Conference on Recent Advancements in Computer, Communication and Computational Sciences (RACCCS-2017) [View project](#)

Multi-Aspect Sentiment Analysis using Domain Ontologies

Srishti Sharma¹, Mala Saraswat², Anil Kumar Dubey³

¹The NorthCap University, Gurugram, India

²School of Computing Science and Engineering, Galgotias University, Greater Noida

³ABES Engineering College, Ghaziabad, U.P

Abstract: Various aspects or characteristic features of an entity come into interplay to create an underlying fabric upon which sentiments blossom. In multi aspect Sentiment Analysis (SA), potentially related aspects of an entity under review are discussed in a single piece of text such as an online review. In this work, we use domain ontologies for enabling multi-aspect Sentiment Analysis. Since, domain ontologies contain the entire domain knowledge, they assist in enhanced aspect identification and detection of the latent or hidden aspects in a review document. We illustrate our approach by developing a system named Ontology driven Multi Aspect Sentiment Analysis (OMASA) system. We provide hotel reviews as input to this system and identify the panorama of explicitly expressed and latent aspects in a review using hotel domain ontology. After detecting the aspects, we link them with the corresponding opinions to gauge the sentiment pertaining to the aspects extracted. OMASA first computes sentiment scores for every aspect of the hotel. It then evaluates the overall sentiment score. On comparing with the baseline, the experimental results of OMASA show a marked improvement in the aspect level evaluation metrics Δ_{aspect}^2 and ρ_{aspect} after detecting the hidden aspects. This shows that OMASA has the potential to identify the latent aspects in text thereby improving the quality of SA.

Keywords: Sentiment Analysis, Domain Ontologies, Machine Learning, Sarcasm, TripAdvisor, Amazon, Twitter.

1. Introduction

There exists abundant literature on successful approaches for document-level sentiment classification [1], [2]. Today, there exist a number of online portals wherein users can pen down their reviews of the merchandise purchased, facilities availed, such as Zomato, Makemytrip. Most of the reviews are self-contained and comprehensive utilized to forecast rankings of merchandise or facilities. E-commerce sites incorporate the feature for reviews and comments as a complementary method of enhancing customer relationship management. Currently, Machine Learning (ML) and neural network driven methods are the contemporary approaches for SA of long documents [3], [4].

However, predicting one sentiment score for each review text is not sufficient, because a review might include different aspects or features of the artefact or amenity under discussion. In the instance of a hotel, different aspects such as “food”, “location” and “service” may be discussed in the document, each having its own significance and bearing its own sentiment.

For example, Fig. 1 shows a sample hotel review. The author reviews the different aspects of a hotel such as hotel location, service, room, etc. These aspects help one tounderstand the major highlights and shortcomings of the hotel. Compared to an overall rating, users may feel encumbered to explicitly rate each and every aspect separately. However, during free flowing writing of reviews and comments, such multi-

aspect sentiments are expressed naturally. Therefore, it is practical to perform multi-aspect SA to forecast distinct scores for every feature rather than ascribe total document score.

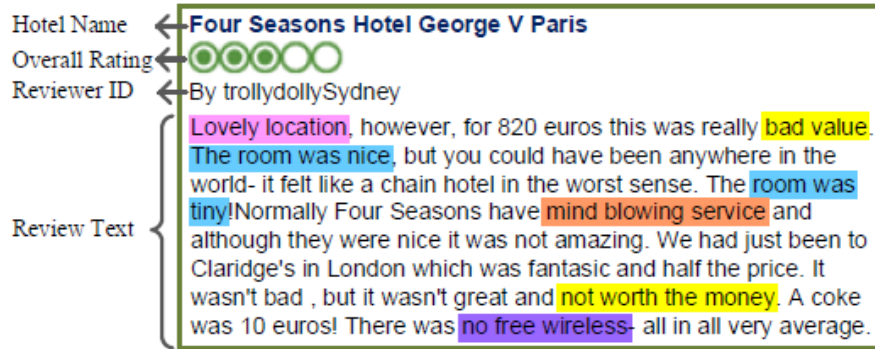


Fig. 1: A sample hotel review

2. Related Work

Some of the earliest works in the area of SA can be traced back to the efforts of Turney et. al [5] in discovering sentiments in product reviews. Around the same time, Pang et al [6] focused on extracting sentiments from movie reviews. In the beginning, SA was mostly limited to document analysis to determine their sentiment polarity to be either positive or negative. Currently, SA can be carried out at the document or review level [5][6][7] at the sentence level [8] and at the feature level [9]. At the document level SA, the complete document's opinion is divided in different sentiment like positive or negative or neutral. In this, we presume that the entire document conveys only a single opinion only, rather than a combination of sentiments. In sentence level SA, we analyze if the given sentence conveys a positive or a negative or a neutral sentiment. At sentence level, we distinguish the objective sentences from the subjective sentences. In this, our main goal is to identify and extract the object feature/aspect about which the sentiment has been expressed and to categorize those sentiments as positive or negative or neutral.

Chakraverty et. al [10] in their paper developed a systematic approach to extract and analyse in real time, the macro-level emotional transitions of a city's inhabitants from their Twitter postings [10]. Saraswat et. al in their research harness reviews as the content generated from user to exploit, emotion as a basis for generating recommendations [11].

Sharma et. al developed a novel SA algorithm utilising the contextual information of the words comprising a document. Using this as a foundation we were able to measure the degree to which the words in a document influence each other and the impact of the dynamics of this relationship on the overall document sentiment [12]

Ontologies are being progressively used to capture the semantics of information from various sources. Saraswat et. al [13] focused on designing and implementing an efficient algorithm that will find similar objects in a given ontology tree using Earth Movers distance using MESH (Medical Subject Heading) ontology and Ohsumed Test Collection.

Xianghua et. al in their paper propose an unsupervised approach to automatically discover the aspects discussed in Chinese social reviews and also the sentiments expressed in different aspects. They apply the Latent Dirichlet Allocation (LDA) model to discover multi-aspect global topics of social reviews, and then extract the local topic and associated sentiment classified by HowNet lexicon [14].

Since domain ontologies contain the entire domain knowledge, they assist in enhanced aspect identification and detection of the latent or hidden aspects in a review document. Different aspects may often be linked to one another and occur together naturally. For instance, the expression, “the chamber was messy” describes the explicitly mentioned feature “chamber”. In addition, it gives an opinion about the independent aspect “cleanliness” which is not explicitly mentioned in the phrase. OMASA has the capability to identify such latent and hidden aspects and mine sentiments from them appropriately, which is not possible by aspect based sentiment analysis approaches proposed earlier in literature.

Proposed approach also incorporates appropriate mechanisms to identify and handle sarcastic text, which may otherwise affect the results of Sentiment Analysis. This is often ignored in SA approaches.

3. Proposed Work

According to [2], an opinion is a quintuple $(o_j, f_{jk}, so_{ijkl}, h_i, t_l)$ where o_j is the target object, f_{jk} is an aspect k of the object j , so_{ijkl} is the sentiment value of the opinion of the opinion holder h_i on aspect f_{jk} of object o_j at time t_l . The sentiment factor so_{ijkl} may be positive, negative, neutral or it may express an extra granulated assessment. In terms of this definition, SA is the problem of estimating correctly the sentiment so_{ijkl} at time l as expressed by the author i on an aspect k of the entity j which is discussed in any opinionated document d . It indicates the opinion of the reviewer regarding a corresponding feature or aspect of the item. We explain the steps of the OMASA approach for Aspect based SA in the following sub-sections.

3.1 Pre-Processing

The OMASA system first converts all the words in the text to lower case to maintain a consistent casing. Next, co-reference resolution is performed. Co-reference resolution is the problem of identifying all the terms that address the same real world entity in a writing [15]. For instance, consider the sentence, “*Eve said Jack was her brother*”. In this sentence, the proper noun *Eve* and the pronoun *her*, both denote the same entity, *Eve*. Similarly, the nouns *Jack* and *brother* refer to the same entity, *Jack*. In this example, *Eve* and *her* represents a pronominal co-reference whereas *Jack* and *brother* represents a nominal co-reference.

In order to arrive at the precise meaning and polarity from a writing, pronouns and other such referring terms need to be linked accurately with the entities that they actually denote. Set of rules aimed at determining the co-references usually look for the previously occurring, bordering entities attuned to the referring term. We perform co-reference resolution using Stanford CoreNLP [16], [17]. Stanford CoreNLP implements pronominal as well as nominal co-reference resolution. The output of the co-reference resolution engine of Stanford CoreNLP is a co-reference graph. We save this co-

reference graph and use it later when the system processes the sentiments associated with all the entities.

3.2 Sarcasm Detection

Content posted by users on the internet such as blogs, microblogs, online reviews, etc., is often loaded with sarcasm. One of the core challenges of SA is to evaluate phrases loaded with sarcasm. Understanding the underlying context of a text is important to understand whether sarcasm is present as a hidden sentiment or not. Hence, it is a challenging assignment in the domain of Natural Language Processing (NLP).

The authors in [18] explored the presence of sarcastic expressions in tweets. In particular, they calculated the effect of sarcasm on SA. Their experiments show that if sarcasm is identified and accounted for appropriately, SA improves by approximately 50% as compared to SA without detecting sarcasm.

3.2.1 Training Corpus

Our training dataset consists of a pre-compiled sarcasm dataset from Twitter [19] and a corpus of Amazon product reviews. The Twitter dataset has a total of 1000 tweets. It is composed of tweets that are author annotated for sarcasm, bearing hashtags #sarcasm or #irony, similar to the approach described in [20]. The corpus of Amazon product reviews contains a total of 1995 Amazon product reviews, taken from the Sarcasm Corpus v1 [21].

3.2.2 Aspect Term Extraction using Domain Ontology

Domain ontologies present domain knowledge in a hierarchical fashion. A domain ontology shows the top level concepts in the domain and illustrates their comprehensive descriptions and interrelationships with other domain concepts. It represents the entities (classes) in a domain, objects, object properties and the links that exist among them. Domain ontologies are a tool that enable collaboration of domain knowledge. Mapping with domain ontologies is carried out to identify the aspects on which a reviewer has commented. Domain ontologies allow for enhanced aspect identification and detection.

Since we are working on the hotel domain, we map with a domain ontology for the hotel domain. However, this is just to illustrate the utility of this method. This may be done for other domains as well like movies or any other products like camera, etc. We map with a hotel domain ontology namely Hontology [22]. Hontology contains overall 282 concepts organized into 16 top-level concepts and sub-concepts. All the concepts and sub-concepts in the ontology have some relevance to the hotel domain, and the occurrence level of a concept in the domain hierarchy is indicative of the concept's relevance to the domain. The depth of Hontology is 5. The top level concepts are Accommodation, Facility, Service, Staff, Room and Guest Type, Design, Meal, Points of Interest, Price, Rating and Staff, etc. Fig. 2 gives a representation of Room types, Service and Staff in Hontology. As we can clearly see, for the top level concept service, there exists a number of sub-concepts like Housekeeping, Airport Shuttle, Babysitting, etc. These sub-concepts also provide us valuable information about how an aspect is discussed by a reviewer. Hence, we also take the sub-concept words of every top level concept into account for aspect identification and detection.

Hontology contains information about 16 concepts corresponding to the hotel domain, however in our hotel review dataset, we have hotel reviews, overall hotel ratings and ratings for 7 aspects of the hotel domain per review. Hence, in this work we evaluate a hotel only on 7 aspects for which we have ground truth ratings. However, using Hontology we could evaluate a hotel on total 16 aspects. We map all the aspects that we need to evaluate on with Hontology concepts. For example, the aspect Rooms is mapped to concepts Room and Accommodation in Hontology, aspect Location is mapped to concepts Location and Points of Interest, aspect Service is mapped to Hontology concepts Service and Staff, aspect Check in/Front Desk is mapped to Hontology concept Reception-Check in service-Check out service, etc. Once an aspect is mapped to a Hontology concept, we extract all the concept and sub-concept words of that particular Hontology concept and add them to the set of aspect keywords corresponding to that particular aspect.

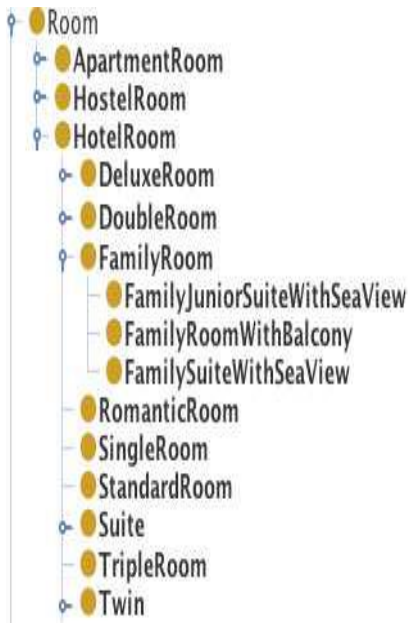


Fig. 2: Representation of Room types, Service and Staff in Hontology

Aspect Term Extraction using Hontology is carried out as explained in process *GetAspectAnnotations(.)* in Fig. 4. In line 1, the input review R to be analyzed is split into its constituent sentences. Starting from line 2, the review is processed sentence-wise. In line 3, $Count(j,i)$ that records the matching hits between sentences and aspects is initialized to zero for each sentence S_j and aspect K_i . The value of i varies from 1 to 7 as there are seven aspects to be evaluated. In line 4, sentence S_j is split into its corresponding token set T_j . In lines 5-7, first of all we compare each token t_z in sentence S_j with the set of aspect keywords K_i corresponding to every aspect a_i . If t_z is in K_i , this is a hit for sentence S_j and aspect a_i . In this case, $Count(j,i)$ is incremented by 1. Otherwise, we compute the semantic similarity between any token t_z in S_j and the all the aspect keywords for each of the aspects a_i . If this similarity for some aspect keyword K_{ni} of aspect a_i comes out to be more than a threshold Θ , we record this as a hit for sentence S_j and aspect a_i and increment $Count(j,i)$ by 1 (lines 8-10). This helps us discover the latent or hidden aspects in a review that a reviewer may have commented on without explicitly using any aspect keyword. We explain the semantic similarity

computation in the next subsection. We determine the threshold Θ experimentally. In line 11, we label a sentence S_j by aspect a_i if $Count(j,i) > 0$. A sentence may be labeled by multiple aspects, indicating that one sentence contains information pertaining to multiple aspects. Finally, in line 13 we output all the sentences in the review annotated with their aspect assignments.

Semantic Similarity Computation

In the *GetAspectAnnotations(.)* algorithm discussed in Fig. 3, we compute the semantic similarity between all the aspect keywords of every aspect and all the words in each sentence of a review. This step helps in discovering the latent or hidden aspects in a review that the reviewer may have commented on without explicitly using any aspect keyword. For example, a hotel review may contain a sentence, “The room was filthy”. In this sentence, there is explicit mention of the aspect room and implicit mention of the aspect cleanliness. The method proposed for Aspect Term Extraction discovers such relationships. For semantic similarity computation, we want a measure reflective of the semantics of a term, rather than its lexical category. For instance, the verb “marry” should be semantically similar to the noun “wife”. Hence, we use Word vector-based similarity or discordance to estimate the semantic similarity. We use two different types of word embedding to capture this similarity, namely:

-GloVe: We utilize the pre-trained vectors of the GloVe project. The lexicon magnitude is 2,195,904 [23].

-Word2Vec: We utilize pre-trained Google word vectors. These were trained using Word2Vec tool on the Google News corpus. The lexis magnitude for Word2Vec is 3,000,000 [24].

GetAspectAnnotations(.)

Input: A review R and a set of aspect keywords $K_i = \{K_{1i}, K_{2i}, \dots, K_{Ni}\}$ corresponding to each aspect a_i in A for $i=1$ to 7

Output: R fragmented into sentences annotated with feature labels.

Begin

1. Fragment R into sentences, $R = \{S_1; S_2; \dots, S_J\}$
2. **For** each sentence S_j in R :
3. Initialize $Count(j,i)=0$ for $i=1$ to 7
4. Split the sentence into a set of unigram tokens $T_j = \{t_1, t_2, \dots, t_z\}$
5. **For** each token t_z in S_j :
6. **If** t_z in K_i :
7. $Count(j,i) = Count(j,i) + 1$
8. **Else If** $sim_{v_n}(t_z, K_{ni}) > \Theta$
9. $Count(j,i) = Count(j,i) + 1$
10. **End for**
11. Assign the sentence S_j an aspect label a_i if $Count(j,i) > 0$. A sentence may be assigned with multiple aspects.
12. **End for**
13. Output the annotated sentences with aspect assignments.

End

Fig. 3: The Aspect Segmentation Algorithm

3.4 Target Opinion Extraction

After identifying the multiple aspects that a reviewer has commented on in a review, the next task is Target Opinion Extraction. This task identifies the opinion corresponding to each target, that is, aspect and estimates its polarity.

Features having both syntactic and semantic content, like dependency trees, frequently exhibit better performance in SA systems [25]. The dependency parser is capable of handling effectively conjunctions, negations and bigram aspect terms. As such, it exhibits the best performance as per [26]. Some sample grammatical relations extracted by the dependency parser are listed in Table 1.

3.4.1 Neighborhood Relation

Any two consecutive words $w_i, w_{i+1} \in S$, if $w_i, w_{i+1} \notin \text{Stopwords}$ in a sentence S in a review R are related by the neighborhood relation. *Stopwords* is a list of words like ‘he’, ‘she’, ‘the’, etc. We ignore these words and do not take them into consideration because they do not contribute to sentiments.

3.4.2 Significant Dependency Relations

Any two words w_i and w_j in a sentence S of review R are related by a dependency relation if $(w_i, w_j) \in \text{Significant_Dependency_Relation}$. We extract Significant Dependency Relations from the dependency parse tree of S . The dependency relations extracted from the dependency parse tree of the statement, “I stayed at the Hilton and its nice but I'm probably the only person who dislikes their garden” are shown in Fig. 5. The set of significant dependency relations listed above is not minimal because not all the relations in the set are equally significant for Target Opinion Extraction. Taking cue from past work as in [27], we illustrate pruning the relation set and obtaining a minimal set of significant relations, by leave-one-relation out test or ablation test on a seed set of data.

3.5 Classification

We represent every sentence S in a review R as a set of vectors V where each vector consists of an aspect a_i in A and its associated opinion words w obtained from SRS and Neighborhood Relation set. These set of vectors then form the input for any supervised classification system.

3 Experiments and Results

We now describe the dataset used, Review level and Aspect level SA results.

3.1 Dataset Used

We use the hotel review dataset of Wang et al. 2010 [28] for our valuation. This dataset encompasses approximately 250,000 periodical evaluations of hotels from the website Tripadvisor that mention about 1850 hotels covering over 60 different sites. For each review there exists text, associated mathematical evaluations and metadata. In this corpus, reviews have overall ratings, and ground truth aspect ratings on 7 aspects: Value, Room, Location, Cleanliness, Check in/front desk, Service, and Business service. All the rankings in this corpus, vary from 1 star to 5 stars.

3.2 Review Level SA Results

We evaluate the reviews in the dataset for overall and aspect wise predictions. For overall review evaluation, we divide all the reviews in this dataset into three classes, that is, Positive, Negative and Neutral. We consider Reviews rated 4-5 as positive, reviews rated 3 as neutral and those rated 1-2 as negative. After this, total 178,487 reviews are positive, 36316 negative and 25262 neutral. Since this data is highly skewed, while training our classifiers we perform random under sampling. We train on 50-80% of the reviews after under sampling in each of the positive, negative and neutral classes. We test on the remaining samples. We illustrate the experimental results by three classifiers Multiclass Support Vector Machine (SVM), Naïve Bayesian (NB) and Maximum Entropy (MaxEnt) in Tables 4-7. After classification, we reverse the polarity of reviews labeled as sarcastic in the test set.

In Table 1 that depicts the results for 80% training and 20% test data, SVM performs the best with 82.18% accuracy. The accuracies obtained by NB and MaxEnt are almost comparable. The TPR values in all cases are significantly higher than the corresponding FPR values. SVM outperforms the other two classifiers in case of all positive, negative and neutral samples in terms of accuracy, FPR and TPR barring FPR for negative samples. FPR for negative samples is least by NB classifier, yielding only 5.94%

Table 1: Review level quantitative evaluation of the performance of different classifiers

Classifier	TPR (in percent)			FPR (in percent)			Accuracy
	Positive	Negative	Neutral	Positive	Negative	Neutral	
Naïve Bayes	80.17	81.32	84.56	8.49	5.94	12.54	80.40
SVM	81.92	83.77	85.21	7.33	6.74	10.47	82.18
Max Entropy	79.73	81.16	83.27	7.82	7.58	12.51	79.96

In Table 2 that depicts the results for 70% training and 30% test data, SVM performs the best with 78.79% accuracy. The TPR values in all cases are significantly higher than the corresponding FPR values. SVM outperforms the other two classifiers in case of all positive, negative and neutral samples in terms of accuracy, FPR as well as TPR.

Table 2: Review level quantitative evaluation of the performance of different classifiers

Classifier	TPR (in percent)			FPR (in percent)			Accuracy
	Positive	Negative	Neutral	Positive	Negative	Neutral	
Naïve Bayes	76.99	79.10	80.01	9.32	9.20	13.42	77.33
SVM	78.22	81.54	84.18	8.10	7.32	12.61	78.79
Max Entropy	76.11	78.73	81.35	10.68	8.38	12.84	76.58

In Table 3 that depicts the results for 60% training and 40% test data, NB classifier performs the best with 76.65% accuracy. SVM is a close second with 76.56% accuracy. The TPR values in all cases are significantly higher than FPR values. SVM outperforms the other two classifiers in case of all positive, negative and neutral samples in terms of accuracy and TPR. FPR values are best for NB classifier for all positive, negative and neutral samples.

Table 3: Review level quantitative evaluation of the performance of different classifiers

Classifier	TPR (in percent)			FPR (in percent)			Accuracy
	Positive	Negative	Neutral	Positive	Negative	Neutral	
Naïve Bayes	74.31	76.02	77.26	0.11	0.10	0.15	76.65
SVM	75.92	79.03	82.26	9.51	9.14	12.75	76.56
Max Entropy	73.37	75.26	78.69	12.43	9.75	14.16	73.85

In Table 4 that depicts the results for 50% training and 50% test data, SVM performs the best with 72.74% accuracy. NB is a close second with 72.17% accuracy. The TPR values in all cases are significantly higher than FPR values. SVM outperforms the other two classifiers in case of all positive, negative and neutral samples in terms of accuracy, FPR as well as TPR barring FPR for negative samples which is least by NB classifier.

Table 4: Review level quantitative evaluation of the performance of different classifiers

Classifier	TPR (in percent)			FPR (in percent)			Accuracy
	Positive	Negative	Neutral	Positive	Negative	Neutral	
Naïve Bayes	69.90	70.27	72.48	13.76	9.94	16.48	72.17
SVM	72.08	75.11	76.94	10.93	13.42	13.58	72.74
Max Entropy	68.68	70.85	73.41	13.25	12.12	18.16	69.23

3.3 Aspect Level SA results

Let $D = \{R_1; R_2; \dots R_j\}$ be a set of review text documents. Given the complete set of review documents D where each review R is associated with I aspects $\{A_1; A_2; \dots A_I\}$ to be analyzed. We use two measures to quantitatively evaluate the performance of different classifiers aspect wise. We explain them as under:

(1) *Mean square error on aspect rating prediction (\mathcal{A}^2_{aspect}):* Suppose f^*R_i is the ground-truth or the actual rating obtained from the training data for aspect A_i in review R . \mathcal{A}^2_{aspect} is indicative of the difference amid the predicted aspect rating fR_i and f^*R_i . It is represented as explained in Equation 4.

$$\mathcal{A}^2_{aspect} = \sum_{R=1}^{|D|} \sum_{i=1}^I (f_{R_i} - f^*_{R_i})^2 / I * |D| \quad (4)$$

Where $|D|$ is the total amount of review documents.

(2) *Aspect correlation inside reviews* (ρ_{aspect}): ρ_{aspect} measures how accurately the expected aspect scores can preserve the comparative order of aspects in a review as indicated by the ground-truth ratings. For instance, in a review, the author may have enjoyed the locality more than the sanitation. The factor ρ_{aspect} measures whether the projected scores retain this inclination. ρ_{aspect} is defined as given by Equation 5.

$$\rho_{aspect} = \sum_{R=1}^{|D|} \rho_{f_R, f_R^*} / |D| \quad (5)$$

where ρ_{f_R, f_R^*} is the Pearson correlation coefficient between two vectors f_R and f_R^* .

Pearson correlation coefficient is used because it is indicative of the link amid two variables. So, this helps to measure the association between the ground truth and predicted ratings. The value of this coefficient varies from -1 to 1. If the assessment is close to 1, it indicates perfect correlation: as one variable surges, the other variable also inclines to surge. If the assessment is close to -1, as one variable surges, the other variable inclines to shrink. If the assessment is zero, it indicates no correlation. The Pearson correlation coefficient amid any two variables x and y is given by r as shown in Equation 6.

$$r_{xy} = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n \sum x_i^2 - (\sum x_i)^2} \sqrt{n \sum y_i^2 - (\sum y_i)^2}} \quad (6)$$

where n represents sample size and x_i, y_i are the distinct sample points indexed with i .

We summarize the aspect level results in Table 8. Though SVM performs much better than NB and MaxEnt classifiers in review level analysis as well as on ρ_{aspect} , it does not perform the best on Δ^2_{aspect} . NB has the best value of 0.575. A high value of ρ_{aspect} indicates that SVM is more superior in differentiating the ratings of diverse aspects inside a review. Such a hypothesis concerning the relative preferences on different aspects cannot be acquired with a single overall rating.

The factor Δ^2_{aspect} indicates the nonconformity of each predicted aspect rating and ground-truth rating independently. Thus, the lesser this value is, the better the classifier. However, Δ^2_{aspect} does not reflect how well the relative order of aspects in a review is preserved. For example, take a case where there are five aspects and a review having an overall rating of 4 and ground-truth aspect ratings of (3; 4; 4; 5; 4). An aspect-unaware prediction of (4; 4; 4; 4; 4), cannot differentiate the different aspects. It has $\Delta^2_{aspect} = 0.67$. It has a ρ_{aspect} of 0. This indicates no correlation between the ground truth and predicted aspect ratings.

Now take another prediction (2; 3; 3; 5; 4), which can tell the real difference between aspects. It has a $\Delta^2_{aspect} = 1$, which is higher, thus showing more difference between predicted and gold standard ratings. This is worse than the aspect unaware prediction. However, in this case the ρ_{aspect} is 0.93 which shows that the aspect ratings order is preserved. We clearly see that even an aspect unaware prediction result can have a lesser individual deviation in aspect ratings from gold standard. We conclude that even though

Δ^2_{aspect} indicates the difference between the predicted and gold standard ratings, it is of little significance without ρ_{aspect} values to correlate along with the relative rating order.

In Table 5 we present the results of the aspect-level quantitative evaluation of the performance of different classifiers by OMASA and baseline. For OMASA, the NB classifier achieves the best Δ^2_{aspect} of 0.575, but since it under-performs on ρ_{aspect} having the lowest value of 0.139, it cannot be considered the best. On the other hand, SVM performs better by having a Δ^2_{aspect} of 0.684 and ρ_{aspect} of 0.472. In the baseline evaluation, we do not carry out hidden aspect detection and significant relation set extraction. Taking a look at Table 8, we can see that ρ_{aspect} is lesser for baseline by each of the three classifiers that we applied. This shows that OMASA is better at preserving the relative order of aspects in the ground-truth ratings than the baseline. On the other hand, the baseline Δ^2_{aspect} values are higher than OMASA in each case which indicates more deviation from ground truth ratings. This clearly establishes the superiority of OMASA over the baseline.

Table 5: Aspect-level quantitative evaluation of the performance of different classifiers

Classifier	Δ^2_{aspect}		ρ_{aspect}	
	OMASA	Baseline	OMASA	Baseline
NB	0.575	0.693	0.139	0.120
SVM	0.684	0.701	0.472	0.411
Max Ent	0.672	0.686	0.294	0.267

4. Conclusions and Future Work

In this work, we used domain ontologies for performing multi-aspect SA of hotel reviews by developing our system OMASA. Since the domain ontology on hotels contain the entire domain knowledge, we were able to improve aspect identification in the corpus of hotel reviews. Specifically, we were able to detect the latent or hidden aspects in the review. We employed dependency parsing and neighborhood relations to extract opinion expressions pertaining to the identified aspects. The system learnt the minimal set of significant relations used by the dependency parser. We performed both document (review) level SA and aspect level SA on these reviews using three different ML algorithms- SVM, NB and MaxEnt. SVM classifier reported the best results amongst the three.

We compared our approach with a baseline that does not have hidden aspect detection and does not learn the significant relations of the dependency parser. The aspect level SA results reported by OMASA were better than the baseline in terms of the evaluation metrics ρ_{aspect} as well as Δ^2_{aspect} . The best value of ρ_{aspect} reported by OMASA is 6.1% more than the best value reported by the baseline, indicating better association between the ground truth and predicted aspect ratings by OMASA. The best value of Δ^2_{aspect} by the baseline is 11.1% higher than OMASA, indicating lesser deviation between ground truth and predicted aspect ratings by OMASA than by the baseline. This establishes the

superiority of OMASA over the baseline. This re-affirms our belief that indeed there are certain latent or hidden aspects present in online reviews and detecting these helps improve the accuracy of SA. In the next chapter, we summarize all our works and present the conclusions.

References

1. B. Pang and L. Lee, "Opinion mining and sentiment analysis," in *Foundation and Trends in Information Retrieval*, 2007, pp. 2(1–2), 1–135
2. B. Liu, "Sentiment Analysis and Subjectivity-Handbook of Natural Language Processing", 2010.
3. D. Tang, B. Qin and T. Liu., "Document modeling with gated recurrent neural network for sentiment classification," *In EMNLP. pages 1422–1432*, 2015.
4. Z. Yang, D. Yang, C. Dyer, X. He, A. Smola and E. Hovy., "Hierarchical attention networks for document classification," in *Proceedings of NAACL-HLT, pages 1480–1489.*, 2016.
5. Turney, P. (2002) 'Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews', *Proceedings of the Association for Computational Linguistics*, pp.417–424.
6. Pang, B., Lee, L. and Vaithyanathan, S. (2002) 'Thumbs up? Sentiment classification using machine learning techniques', *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp.79–86.
7. Gräbner, D., Zanker, M., Fliedl, G. and Fuchs, M. (2012) 'Classification of customer reviews based on sentiment analysis', *19th Conference on Information and Communication Technologies in Tourism (ENTER)*, Springer, Helsingborg, Sweden.
8. Celikyilmaz, A., Hakkani-Tur, D. and Feng, J. (2010) 'Probabilistic model-based sentiment analysis of Twitter messages', *The 2010 IEEE International Conference on Spoken Language Technology, Workshop*, pp.79–84.
9. Singh, V.K., Piryani, R., Uddin, A. and Waila, P. (2013) 'Sentiment analysis of movie reviews', *Proceedings of International Multi Conference on Automation, Computing, Control, Communication, and Compressed Sensing*, pp.712–717, Kerala, India.
10. Chakraverty, S., Sharma, S., & Bhalla, I. (2015). Emotion–location mapping and analysis using twitter. *Journal of Information & Knowledge Management*, 14(03), 1550022.
11. Saraswat, M., Chakraverty, S. (2021). Emotion Distribution Profile for Movies Recommender Systems. In: Sharma, H., Gupta, M.K., Tomar, G.S., Lipo, W. (eds) *Communication and Intelligent Systems. Lecture Notes in Networks and Systems*, vol 204. Springer, Singapore. https://doi.org/10.1007/978-981-16-1089-9_30
12. Sharma, S., Chakraverty, S., Sharma, A., & Kaur, J. (2017). A context-based algorithm for sentiment analysis. *Int. J. Comput. Vis. Robotics*, 7(5), 558-573.
13. Saraswat, M. (2010, August). Efficiently Finding Similar Objects on Ontologies Using Earth Mover's Distance. In *International Conference on Database and Expert Systems Applications* (pp. 360-374). Springer, Berlin, Heidelberg.
14. Xianghua, F., Guo, L., Yanyan, G., & Zhiqiang, W. (2013). Multi-aspect sentiment analysis for Chinese online social reviews based on topic modeling and HowNet lexicon. *Knowledge-Based Systems*, 37, 186-195.
15. C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard and D. McClosky, "The Stanford CoreNLP Natural Language Processing Toolkit," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 55-60, 2014.

16. K. Clark and C. D. Manning, "Deep Reinforcement Learning for Mention-Ranking Coreference Models," in *Proceedings of EMNLP*, 2016.
17. D. Maynard and M. A. Greenwood, "Who cares about sarcastic tweets? Investigating the impact of sarcasm on sentiment analysis," in *Lrec*, 4238-4243, 2014.
18. Srishti Sharma and Shampa Chakraverty, "Sarcasm Detection in Online Review Text", *ICTACT JOURNAL ON SOFT COMPUTING*, APRIL 2018, VOLUME: 08, ISSUE: 03
19. "Data Mining Project focused on Twitter Sarcasm Measurement," [Online]. Available: <https://github.com/dmitryvinn/twitter-sarcasm-measurement>.
20. C. Liebrecht, F. Kunneman and A. Van den Bosch, "The perfect solution for detecting sarcasm in tweets #not", In *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*,, 2013.
21. S. Lukin and M. Walker, "Really? Well. Apparently Bootstrapping Improves the Performance of Sarcasm and Nastiness Classifiers for Online Dialogue," in *The Workshop on Language Analysis in Social Media (LASM), at The Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, Atlanta, Georgia, USA, 2013.
22. M. S. Chaves, L. Freitas and R. Vieira, "Hontology: A Multilingual Ontology for the Accommodation Sector in the Tourism Industry," in *KEOD2012-InternationalConferenceonKnowledgeEngineeringandOntologyDevelopment*, pp: 149-154, 2012.
23. J. Pennington, R. Socher and C. D. Manning, "GloVe: Global Vectors for Word Representation," 2014.
24. T. Mikolov, K. Chen, G. Corrado, J. Dean, "Efficient Estimation of Word Representations in Vector Space," CA, USA, 2013.
25. T. Nakagawa, K. Inui and S. Kurohashi, "Dependency Tree-based Sentiment Classification using CRFs with Hidden Variables," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL*, pages 786–794, Los Angeles.
26. S. Moghaddam and M. Ester, "On the Design of LDA Models for Aspect-based Opinion Mining," in *CIKM'12*, Maui, HI, USA, October 29–November 2, 2012.
27. S. Mukherjee and P. Bhattacharyya, "Feature Specific Sentiment Analysis for Product Reviews," in *CiCLing 2012*.
28. H. Wang, Y. Lu and C. Zhai, "Latent Aspect Rating Analysis on Review Text Data: A Rating Regression Approach," in *The 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'2010)*, p783-792. , 2010.