Summarizing Customer Reviews through Aspects and Contexts

Prakhar Gupta[1], Sandeep Kumar[1], and Kokil Jaidka[2]

[1] Department of Computer Science and Engineering, IIT Roorkee, Roorkee, India
prakharguptage@gmail.com,
sgargfec@iitr.ac.in
[2] Adobe Research India Labs, Bangalore, Karnataka, India
koki0001@e.ntu.edu.sg

Abstract. This study leverages the syntactic, semantic and contextual features of online hotel and restaurant reviews to extract information aspects and summarize them into meaningful feature groups. We have designed a set of syntactic rules to extract aspects and their descriptors. Further, we test the precision of a modified algorithm for clustering aspects into closely related feature groups, on a dataset provided by Yelp.com. Our method uses a combination of semantic similarity methods- distributional similarity, co-occurrence and knowledge base based similarity, and performs better than two state-of-the-art approaches. It is shown that opinion words and the context provided by them can prove to be good features for measuring the semantic similarity and relationship of their product features. Our approach successfully generates thematic aspect groups about food quality, décor and service quality.

Keywords: Aspect Detection, Text Classification, Clustering, Text Analysis, Information Retrieval, Opinion Mining, Online Reviews.

1 Introduction

Online reviews are an important resource for people, looking to make buying decisions, or searching for information and recommendations about a product or business. Online review websites like Yelp provide a way for information seekers to browse user reviews, ratings and opinions about the different aspects of service at restaurants and hotels. However, sifting through a large number of reviews to understand the general opinion about a single aspect, is a tedious task. This is the research problem addressed in approaches for aspect mining and analysis, where the aim is to automatically analyze user reviews and generate a summary around the various aspects of a product.

The approach followed in aspect mining studies is to extract parts of speech or aspect-sentiment pairs [1]. In the current work, we extract aspects-descriptor pairs through the syntactic, contextual and semantic features of text, and cluster them into meaningful, related feature groups. People also tend to mention their thoughts about related aspects in the same sentence, which can be leveraged to provide context for

A. Gelbukh (Ed.): CICLing 2015, Part II, LNCS 9042, pp. 241–256, 2015.
DOI: 10.1007/978-3-319-18117-2_18

aspects. The context provided by words such as "delicious" and "uncooked" can prove to be a good indicator of the category of the aspect (in this case, "food") they are used with. Using sources such as WordNet [2] can further help to relate similar aspects – for example, "water" is related to "drink", and "pasta" and "italian pasta" should belong to the same group. Together, these features comprise the heart of our aspect clustering method.

The application of this work is in summarizing a large set of reviews around the aspects they comprise. There are two major contributions of this work-

1. A set of syntactic rules to find aspects, and their opinion carrying descriptors, within sentences of reviews.
2. A clustering algorithm for identifying and clustering similar aspects, using similarity features based on context and thesauri.

This paper is organized as follows: Section 2 describes the related work. Section 3 discusses our problem statement in detail. Section 4 presents the methodology. Section 5 gives the experiments and results. Section 6 discusses results and Section 7 contain conclusions and future works.

2 Related Work

Work in aspect detection is diverse, and syntactic approaches [3] are as popular as knowledge-rich approaches [1]. Several studies have focused on extracting aspects along with their opinions by using dependency parsing [3] or relationships between noun and verb phrases [6]. Hu and Liu [4] and Yi and Niblack in [5] extracted aspect as noun phrases, by using association rule mining and a set of aspect extraction rules and selection algorithms respectively; however, these methods did not perform well with low frequency aspects, such as specific dishes in a restaurant. Several studies have focused on extracting aspects along with their opinions - [3] used dependency parsing to find relationships between opinion words and target expressions, and [6] identified noun and verb phrases as aspect and opinion expressions. These ideas motivated our approach for developing syntactic rules for extracting aspects and their describing adjectives and other parts of speech.

Clustering similarity measures may rely on pre-existing knowledge resources like WordNet [1][7]. Popular similarity metrics include Cosine function, Jaccard Index and PMI (Pointwise Mutual Information) to calculate similarity between words. The method proposed by [8] mapped feature expressions to a given domain product feature taxonomy, using lexical similarity metrics. In [9], a latent semantic association model is used to group words into a set of concepts according to their context documents and then they categorize product features according to their latent semantic structures. The authors in [10] grouped words using a graph-based algorithm based on PMI or Chi-squared test. Knowledge-based approaches have usually showed increased precision but lower recall compared to previous work; furthermore, they are also not able to handle cases where the knowledge bases do not contain domain specific knowledge, or do not use word distribution information. In this work, we have tested our own clustering algorithm against the state-of-the-art, MCL clustering algorithm, and compared the results.

Probabilistic approaches for summarizing reviews include applied topic modeling to identify major themes. However, according to Blei et al. [11], topic models like LDA are not suitable for aspect detection in reviews, as they capture global topics, rather than aspects mentioned in the review. Nevertheless, several significant works have aimed to overcome this problem, notably the multi-grain topic model MG-LDA [12][13] was constructed, which attempts to capture global and local topics, where the local topics correspond to aspects. More recent approaches include creation of hierarchies of aspects [14], extraction of aspects using word frequency and syntactic patterns [15], and semi-supervised methods [16]. We have also used one variant of LDA [11] described in section 5 as a baseline for comparing topic models and clustered aspects.

3 Problem Description

Our first research objective is to extract aspects and the words used to describe them.

Task 1- *Extract the aspects and their descriptors.*

Definition (Descriptor) - A word, especially an adjective or any other modifier used attributively, which restricts or adds to the sense of a head noun. Descriptors express opinions and sentiments about an aspect, which can be further used in generation of summaries for the aspects.

Definition (Aspect-Descriptor Pair) - An aspect-descriptor pair consists of an aspect and the descriptor of that aspect. e.g. (sandwich, tasty) in "This is a tasty sandwich".

Sometimes there may be more than one aspect-descriptor pair in a sentence for the same aspect if more than one descriptor is present. In some cases, the descriptor may not modify the aspect directly but may modify the verb, any adjective or any other descriptor of the aspect. In such cases, a separate pair is created for that descriptor and the aspect (e.g. waiter, angry) in "The *waiter* looked at me *angrily"*.

Task 2- *Clustering of aspects into natural groups.*

The next tasks is to group aspects which fall into natural groups; for example, in restaurant reviews, natural groups of aspects may be about food, some particular type of food like Chinese, décor of restaurant etc. This is done by aggregating aspects based on their term similarity, and then using the following features for clustering the aspects and their descriptors-

- Context or co-occurrence of aspects
- External knowledge base based similarity
- Semantic similarity based on aspects' descriptors

4 The Methodology

The proposed framework for our method is shown in Figure 1. It comprises the detailed workflow for the two main research objectives - Discovery of aspect-descriptor pairs and clustering of discovered aspects.

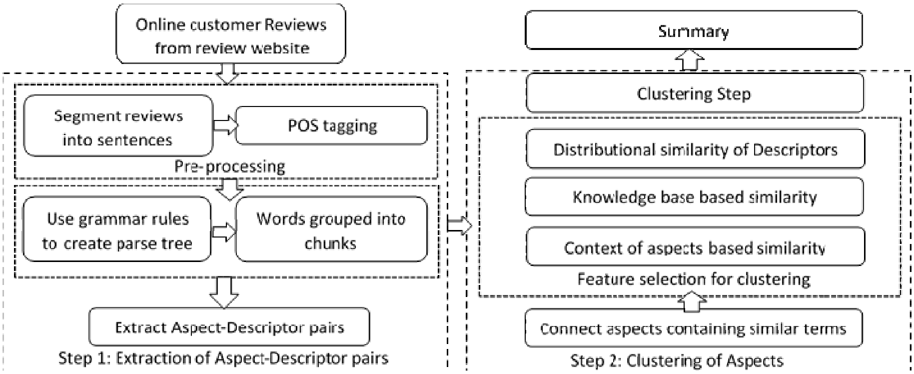

Fig. 1. Framework for aspect-descriptor extraction and clustering

4.1 Extraction of Aspect-Descriptor Pairs

A set of syntactic rules for identifying aspects and descriptors was developed, based on the following observations of the English reviews on Yelp:

- Adjectives, Participles, Articles, Possessive Pronouns and Prepositional Phrases can describe, modify or pointed to a noun.
- Words which describe an aspect are mostly modifiers like adjectives and participles. Adjectives modify or describe a noun. Sometimes Adjectives precede the noun they modify (e.g. "I like the *spicy* burger"), or they may follow a linking verb (e.g. "The burger was *spicy*.").
- Participles are verb forms that can be used as adjectives. An example of Past participles as descriptor is "*bored"* in "I was *bored* at the theatre." Present participle as descriptors can occur in front of a noun (e.g. "I like the *sizzling* dish.") or otherwise (e.g. "I like the dish *sizzling*.").
- Sometimes adverbs also carry information regarding an aspect (e.g. "The restaurant is open *daily*.").

Table 1 shows the custom syntactic rules we have created to identify and extract aspects and their descriptors. The first column provides the "chunk labels" we use to identify the type of aspect-descriptor pair created. In the second column, words in '<>' represent a tag, '<tag.*>' represent a tag can be followed by other letters like VBZ for VB, '<>*' represents zero or more occurrence, '<>+' represents one or more occurrence '?' represents zero or one occurrence. The rules are applied in the sequence given in the table so that a rule with higher priority is detected before a lower priority rule.

Before the extraction of aspect-descriptors pairs, reviews undergo certain pre-processing steps. First, each is review document is segmented into sentences. Next, the reviews are tokenized and lemmatized and parts of speech tagged are obtained, using the Stanford POS tagger, and further transformed into parse trees with the help of Stanford chunker. The result is a tree with words grouped into syntactic, labelled chunks, as in the first column in Table 1. Finally, these are passed to a function which then extracts Aspect-Descriptor pairs from the structures. There may be more than one chunk present in a sentence. A processed chunk can be a used in processing of another rule and become a sub-part of another chunk. If no rule is recognized and still descriptor are present in the sentence (e.g. “It was elegant.”), then the aspect-descriptor pair (“forbusiness”, “elegant”) is generated. Such descriptors are assumed to describe the whole business entity as the aspect.

Table 1. Grammar Rules with examples

<table>
<tr><th>Chunk labels</th><th>Rule*</th><th>Pair extracted</th><th>Example</th></tr>
<tr><td>A</td><td>{<JJ>*<VJ><JJ>*<RB>*<IN|DT>*<NP>}</td><td>(NP,JJ/VJ)</td><td>They have broken windows.</td></tr>
<tr><td>B</td><td>{<JJ>+<CC>?<JJ>?<RB>*<IN|DT>*<NP>
<IN|CC|DT>?<NP>?}/{<JJ><VJ>}</td><td>(NP.JJ)</td><td><i>Dirty</i> and <i>wet</i> bedsheets were found in the room</td></tr>
<tr><td>C</td><td>{<NP>+<W.*|PRP>*<VB.*|VJ><RB>+
<DT>?<JJ|VJ>}</td><td>(NP,JJ/RB)</td><td>Opening is <i>always hectic</i>.</td></tr>
<tr><td>D</td><td>{<B|A>+<VB*|VJ><DT>?<JJ|VJ|RB>*}</td><td>(B|A,JJ/VJ)</td><td><i>Hot</i> sizzler is <i>amazing</i>.</td></tr>
<tr><td>E</td><td>{<NP>+<W.*|PRP>*<VB.*|VJ><DT>?
<JJ|VJ>+}</td><td>(NP,JJ/VJ)</td><td>Rooms are <i>clean</i>.</td></tr>
<tr><td>F</td><td>{<NP.*><W.*>*<VB.*><DT>?<RB>?
<B|A>+}</td><td>(NP,B|A)</td><td>Weekends are <i>great for people</i></td></tr>
<tr><td>G</td><td>{<NP>+<W.*>*<RBR|RBS>?<JJ|VJ></td><td>(NP,JJ/VJ)</td><td>I liked the fish <i>fried</i>.</td></tr>
</table>

**Where JJ are* Adjectives; *VJ ->{<VBG|VBN>*}* are Participle verbs; *RB* are Adverbs; *VB* are Verbs; *NBAR ->{<NN.*|JJ>*} }<JJ*> are nouns and nouns with adjectives*; *NP->{<NBAR><CC|IN><NBAR>}/{<NBAR>}* are noun phrases and noun phrases with conjunctions.

4.2 Clustering of Aspects

For generating aspect-based summaries, aspects which contain the similar terms are aggregated; then, feature values are calculated for every pair of aspects. After calculating the features, the aspects are clustered based on the calculated values. These steps are described in detail below.

Step 1- Connect Aspects Containing Similar Terms

In this step, aspects which are exactly similar or are almost exactly similar in case of multigrams are aggregated into a list of similar aspects, or an aspect-set. It is based on the fact that aspects sharing some words are likely to belong to the same cluster, for example “pool table” and “wooden pool table” most likely refer to the same aspect. Unigrams aspects, which are already lemmatized, are added only to a list of aspect which contain the exact same unigram. For multigrams, an approximate string matching is used. To paraphrase, for every new aspect from list of aspect descriptor pairs, if the incoming aspect is a multigram, say x, its term similarity is first measured against a list of multigrams aspects and if the similarity with another multigram aspect, say y, comes out to be greater than a threshold value, then the multigram aspect x is added to the list of the multigram aspect y, otherwise a new list is initialized with x. The similarity metric used is Jaccard similarity coefficient, eq. 1.

$$Similarity = s(C_i, C_j) = \frac{|s(C_i) \cap s(C_j)|}{|s(C_i) \cup s(C_j)|} \quad (1)$$

Here, $S(C_i)$ represents set of words in string C_i.

Multigrams are not grouped with unigrams because in some cases, if a multigram aspect approximately matching the unigram aspect is added to the set of unigram, then all new incoming unigrams which match with any word of the multigram will be added to the set. For example, if the multigram "poker face" is initialized as an aspect list, then both unrelated unigram aspects "poker" and "face" will be added to the same set.

Step 2- Feature Value Calculation for Clustering
Context or Co-occurrence of Aspects

In this step, aspects which are used in the same contexts are aggregated together based on their co-occurrence patterns, from the following observations of data:

- In general, a sentence is a collection of related aspects with a single focus, which creates a rough semantic boundary. Thus, word distribution information of aspects can be leveraged to cluster aspects based on their context similarity.
- People often express aspects and opinions about them in the same or repetitive syntactic structure, within a single sentence. For example, people often express their opinions about various dishes they ate, their experience with staff, etc. in the same sentence. For example "The fish was tasty but the chicken was overcooked." and "The waiter were friendly and the manager was understanding".
- Unrelated sentences can be a part of same review; but in the sentences itself, related aspects are usually mentioned together.

To gather context information, for every sentence in every review, a context vector is created which comprises all the aspects in the sentence. PMI (or Pointwise Mutual Information) measures the strength of association between two words by comparing the pair of words' bigram frequency to the unigram frequencies of the individual words. It is an indicator of collocation between the terms.

$$PMI(x, y) = \log\left(\frac{p(x, y)}{p(x).p(y)}\right) \quad (2)$$

It has been noticed that bigrams with low frequency constituents may gain high PMI value even when their occurrence probability is low. This problem can be addressed by multiplying the PMI value by an additional term of log of bigram frequency of *x* and *y*, *bigramfreq(x,y),* as suggested in [18]. The final PMI value is given in equation 3.

$$PMI(x, y) = \log\big(bigramfreq(x, y)\big) * \log\left(\frac{p(x, y)}{p(x).p(y)}\right) \quad (3)$$

PMI scores are calculated for each pair of aspect-set and saved to a record. The PMI values of multi-gram aspects is taken as the average value of PMI values of every combination of unigram aspect terms present in the multi-gram aspects. We also incorporate co-occurrence pattern information from this step to group together pairs of co-occurring unigram sets into multigrams. Furthermore, at this stage, pairs which have either low individual probabilities or low PMI (below a threshold) are removed from the record and from the clustering procedure.

External or Knowledge Base Based Similarity

In the next step, given two aspect terms, a1 and a2, we need to find their semantic similarity. We followed a WordNet based similarity approach similar to [19], to take into account both the minimum path length and the depth of the hierarchy path, so that specialized words with concrete semantics are grouped closer together than words in the upper levels of the hierarchy [19] as described in equation 4.

$$\mathrm{sw}(a_1, a_2) = e^{-\alpha l}.\frac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}} \tag{4}$$

Here, α and β are parameters which scale the contribution of shortest path length and depth respectively and their optimal values depend on the knowledge base used, which in case of WordNet is proposed as 0.2 and 0.45 [19]. For the first term in the equation which is a function of path length, the value of l is 0 if both aspects are in the same synsets and are synonyms. If they are not in the same synset but their synsets contains one or more common words, value of l is 1. In all other cases, the value of l is the actual path length between the aspects. For the function of depth, due to the reason explained above, the function scales down the similarity for subsuming words at upper layers and scales it up for subsuming words at the lower layers.

The final similarity score is a value between 0 and 1. In case aspect is not present in WordNet, this value of the feature is taken as 0. If a similarity score is found, then if the pair has a minimum support in the corpus, the value is used in clustering. The similarity values of multi-gram aspects is taken as the average value of similarity values of every combination of unigram aspect terms present in the multi-gram aspects.

Distributional Similarity of Descriptors

Descriptors of an aspect contain semantic information that reflects the relationship of the aspects. Using the similarity of the virtual contexts provided by the descriptors, we can find similar aspects which may not themselves co-occur in the same contexts. Such information can capture implicit aspects which are not evident in the reviews by other features.

We model the semantic similarity of aspects as a function of semantic similarity of their descriptor words, by using a metric of word to word similarity of descriptors which indicates the semantic similarity of the two input aspects. Suppose we have a set for both aspects consisting of their descriptors of the form $S_1= \{d_{11}, d_{12},.., d_{1m}\}$ and $S_2=\{d_{21}, d_{22},\ldots, d_{2n}\}$, where d_{ij} are descriptor words. We first calculate semantic similarity of every pair of descriptors in both set. For semantic similarity of descriptor words, we use the metric normalized PMI as it indicates the degree of statistical dependence between two words. NPMI in equation 5 gives the semantic similarity of two descriptors x and y based of their occurrence and co-occurrence probabilities in the dataset.

$$NPMI(x, y) = \log\left(\frac{p(x, y)}{p(x).p(y)}\right)/p(x, y) \tag{5}$$

We have used another corpus based metric called inverse document frequency which was first introduced by Jones [20]. It is calculated as the log of total number of aspects divided by the number of aspects for which the descriptor is used, which is log(N/n_s). It is based on the fact that descriptors which occur with few aspects contain a greater amount of discriminatory ability than the descriptors that occur with many aspects in the data with a high frequency. This is because such aspects have meaning which relate to particular type of aspects like "delicious" relate to food related aspects.

This metric works well for similar aspects like "decor", "ambiance", "decoration", "furnishing" etc., their descriptors often share the same words like "colorful", "elegant", "modern", "sophisticated" etc. However, descriptors like "great" and "awesome" can be used with a large variety of aspects. Although such descriptors will get low *idf* values, we have manually created a list of very common descriptors which are not included in the calculation of similarity values of aspects. Once we have pairwise similarity values of descriptors from equation 5, they are used to calculate the similarity value of the aspects using the equation 6.

$$sim(A_1, A_2) = \frac{1}{2}\left(\frac{\sum_{d \in A_1}(maxSim(d, A_2) * log(N/n_d))}{\sum_{d \in A_1} log(N/n_d)} + \frac{\sum_{d \in A_2}(maxSim(d, A_1) * log(N/n_d))}{\sum_{d \in A_2} log(N/n_d)}\right) \tag{6}$$

Here, A_1 and A_2 are aspects, d is a descriptor, N is the total number of aspects in the corpus and n_d is the number of aspects d appears with. For each descriptor d in the aspect A1, we identify the descriptor in the aspect A2 with which it gets the maximum similarity value using equation 5. The equation is inspired by work in [21]. The descriptor similarities are weighted with the corresponding inverse document frequencies, then summed up, and an average is taken with the value we get by repeating the same procedure with descriptors of aspect A2. The final value *sim*(A_1, A_2) is an estimate of the similarity between the aspects.

Step 3- Clustering step

Once the above similarity metrics are calculated, aspects are clustered together. We have used two approaches for clustering, one is our algorithm, which we call **Simset clustering** and other is a graph based algorithm called Markov Clustering (MCL) [22]**.** For both algorithms, first, the values of the 3 features described in section 4.2 are calculated; then, a graph is created with aspects as nodes and the value of the features as weight of edges between them. Both algorithms do not require a predefined number of clusters.

In Simset algorithm, first, the aspects are sorted by the total sum of their PMI values with other aspects. Then, for every aspect, a set called simset is initialized which will contain aspects similar to it. Then similarity with every other aspect is measured and if the similarity with aspect a_j is greater than a pre-fixed threshold, for any of the 3 features, then the a_j is either added to the simset of a_i if a_i has not been clustered earlier, otherwise the a_i is added to the simset of a_j and every element a_k in simset of a_i which has similarity value greater than any threshold with a_j, is moved to simset of a_j.

The major difference between Simset and MCL is that in MCL, the edges between each pair of aspects is weighted by their similarity values while in Simset, an edge is present between a pair of aspects only if the similarity value between them is greater than threshold values and the weight is same for each edge.

The MCL algorithm takes the graph as input, and simulates random walk through the graph. It is based on the principle that transition from one node to another within a cluster is much more likely than those in different clusters, and takes into account the weight of their links. The clustering process proceeds in an iterative manner and consists of two steps, one called *expansion,* which corresponds to computing random walks of higher length, which means random walks with many steps and the other called *inflation*, which boosts the probabilities of intra-cluster walks and demotes inter-cluster walks. Increasing the inflation parameter produces more fine-grained clustering. The algorithm converges after some iterations and results in the separation of the graph into different clusters.

Algorithm 1. Simset Clustering of Aspects

for a_i in $A=\{a_1, a_2, \ldots, a_n\}$ **do**
  initialize simset(a_i)
  for a_j in $\{a_{i+1}, a_{i+2}, \ldots, a_n\}$ **do**
    if similarity(a_i,a_j) > any(threshold$_h$) (h={1,2,3}) **then**
      if a_j not already clustered with another aspect **then**
        add a_j to simset(a_i), remove a_j from A
      else add a_i to simset(a_j)
        for all $a_k \in$simset(a_i) **do**
          if similarity(a_k,a_j) > any(threshold$_h$) **then**
            move a_k to simset(a_j)
   else : create simset(a_i) as a new cluster

5 Experiments

This section evaluates the proposed algorithm using dataset obtained from yelp.com. We analyze the performance of Aspect-Descriptor extraction and clustering in detail.

5.1 Dataset Description

Our dataset consists of online reviews of businesses provided by Yelp for "Yelp Dataset Challenge 2014[1]". The dataset consisted of 1,125,458 reviews of different businesses. The reviews were aggregated for every business id and filtered for a minimum number of reviews per business id. Then among the remaining businesses, the reviews of one hotel related business was taken as the final dataset, as it contained a lot of diverse aspects which could be detected and clustered by our approach. The reviews were segmented into sentences giving a total of 6,784 sentences.

---

1 Yelp (Dataset Challenge) `http://www.yelp.com/dataset_challenge`.

5.2 Qualitative Comparison

We show the top 10 clusters (according to size of clusters) detected from our model, in Table 2. The aspects which do not belong to the cluster are struck through. We have also identified a list of descriptor words for every aspect in every cluster in the output. It can be seen that the clusters extracted by Simset are descriptive and informative. Clusters have been manually assigned a label, provided in the first column of the table, and can easily be distinguished as related to service, art, parking etc. For example for the aspect "bed", the list of descriptor words is ["comfort", "super comfi", etc.].

Table 2. Top aspect clusters detected by Simset clustering

Cluster label	Aspects terms	Common Descriptors
Parking related	parking, spot, level, parking garage, park space, fixture parking, light space, spot strip, slot, parking tip, self parking system, plenty of parking, light spot, live space, foot of space, light parking lot, garage, light bulb, ~~reaction, bond~~, flight, garage with number, ~~slot machine, lot of thought , lot of people, lot of restaurant, classic atmosphere lot~~, light alert, support, lot of celeb	Easy, good, easy to find, underground, plenty, open, free, biggest, available
Hotel related	hotel, casino resort, attractive hotel , hotel group, type of hotel, technology hotel, cosmopolitan hotel, detailed hotel, time hotel, thing hotel, genre of hotel, end hotel, boutique hotel, part hotel	Beautiful, amazing, sophisticated
Casino related	casino lounge, casino floor property, floor, ~~2nd floor~~, 3rd floor, ~~ground floor, floor balcony~~, three floor, casino/hotel, casino strip, lobby and casino, local casino, elevator, shelf liquor, casino tour, casino tour, background for picture, ~~entrance,~~ hotel/casino , level casino, casino area, control decade, casino/resort, store sell	Hippest, favorite, modern, unique, new, amazing, open
Night-club marque related	identity club, marque club, ~~card time, card key,~~ credit card, marque management, ~~Identity, identity reward program, henry,~~ one word club, marque nightclub, ~~gold card status~~, glitch on check, pocket marque, line at marque, douchebag club, club downstair, sign marque, 2 night	hot, new, good music, popular, breathtaking
Bar related	chandelier, bartender, bar, casino bar, bar option, time bartend, bartender service, chandelier bar, mini-bar, mini bar space, bar in paris, casino bar restaurant, care minibar, bar sip, ~~pizzeria~~, lobby bar, bar and food event, bond bar, bar stool, chandelier middle, buffet, ~~fridge,~~ bond, vesper, spoon buffet, ~~kink buffet,~~ crystal chandelier, ~~minifridge~~	Crystal, sparkling, marvelous, multi- level, massive
Bath-room related	bathtub knob, deck shower, shower and tub, soak tub, shower area, shower pool, tub, bathroom, sink tub, tub outside, roll of toilet paper, toiletries, hair, bath, ~~bottle of water,~~ bathroom toiletries, bathroom 2, water pressure, ~~check in~~, whirlpool tub, ~~hallway with north~~	Huge, chic, cool, spacious, open
Art related	art, piece, piece of art, artwork, art from artist world, art work, art book, column of art, ~~pong and sport~~, graffiti artist, art display, ~~sport book~~, ~~foosball~~, ~~ping pong~~	Interesting, modern,original, great

Table 2. *(Continued)*

Art related	art, piece, piece of art, artwork, art from artist world, art work, art book, column of art, ~~pong and sport~~, graffiti artist, art display, ~~sport book~~, ~~foosball~~, ~~ping pong~~	Interesting, mod-ern,original, great
Pool related	place, boulevard, feel place, ~~table~~, issue, feel, pool table, pool deck, pool experience, ~~end place~~, ~~room area~~, thing place, ~~bedside table~~, area level, ~~town~~, pool with cabana, ~~pizza place~~, pool day, pool ideal	Beautiful, amazing, mo-dern, edgeless
Cus-tomer service related	cocktail service, week customer service line, customer service, cock-tail waitress, beverage service, room service, service waiting, food from room service, desk service, notch service, player club service, ~~gambler~~, factor and service, ~~internet service~~, food and service, service before	Good, quick, friendly, horrible, poor, great, terrible
Room related	suit, terrace, one bedroom, room terrace studio, terrace suit, bed room studio, ~~one person~~, one bedroomsuite, ~~one thing, one complaint~~, com-fort tower suit, city suit, conference center	Expensive, impressive, special, view

The top 10 clusters from DLDA are provided in Table 3. For DLDA, data was clustered by keeping n=10, and α and β to their proposed values. It is evident from the comparison of two tables that our system gives a better understanding of the aspects. The most reviewed aspects like the nightclub, bar and Casino are also detected explicitly, unlike the DLDA model. In the results of the DLDA model, since all words are unigrams, some of the aspects do not make sense like “lot” in “parking lot”.

Table 3. Clusters detected from DLDA

Topic 1	room, service, time, desk, check, hour, customer, minute, glass, anything, money, coffee, security, employee, food, identity, call, cosmopolitan, gambling, plenty
Topic 2	bar, chandelier, drink, cosmo, friend, crystal, review, elevator, music, ~~design, story, end, top,~~ vibe, ~~desk, art, work, bag, touch~~, ceiling
Topic 3	night, day, club, marquee, property, ~~line, hour~~, party, crowd, weekend, ~~guest, front, way,~~ middle, ~~window, shop, name,~~ woman, ~~course, anyone~~
Topic 4	~~casino~~, lot, parking, lobby, spot, level, space, light, door, shower, garage, ma-chine, ~~thing, art~~, someone, slot, valet, wall~~, three, part~~
Topic 5	place, ~~everything~~, food, stay, card, star, year, guest, book, ~~eye~~, ~~pizza,~~ morning, couple, access, trip, ~~guy, mini, player~~, choice, ok
Topic 6	pool, ~~floor, area~~, restaurant, casino, table, lounge, game, chair, wow, bartender, ~~boulevard, living,~~ atmosphere, box, seat, nightclub, movie, week
Topic 7	room, ~~one,~~ tv, bathroom, bed~~, two,~~ bedroom, ~~friend,~~ screen, ~~side, star, tub~~, fridge, kitchen, ~~time, city, size, idea, part, rest~~
Topic 8	strip, view, suite, balcony, terrace, ~~one, bellagio~~, hotel, ~~something~~, fountain, tow-er, ~~point, phone,~~ people, show, ~~kind~~, building, ~~wrap, word~~
Topic 9	~~hotel,~~ people, ~~cosmo, thing, decor,~~ staff~~, buffet, fun, way, bit, experience, time, spoon, weekend,~~ person, conference, ~~water, system, entrance~~
Topic 10	hotel, vega, cosmopolitan, la, hip, ~~nothing,~~ reason, ~~spa, detail, service,~~ aria, re-view, ~~select, thing,~~ center, ~~note, everyone, cocktail, charge,~~ event

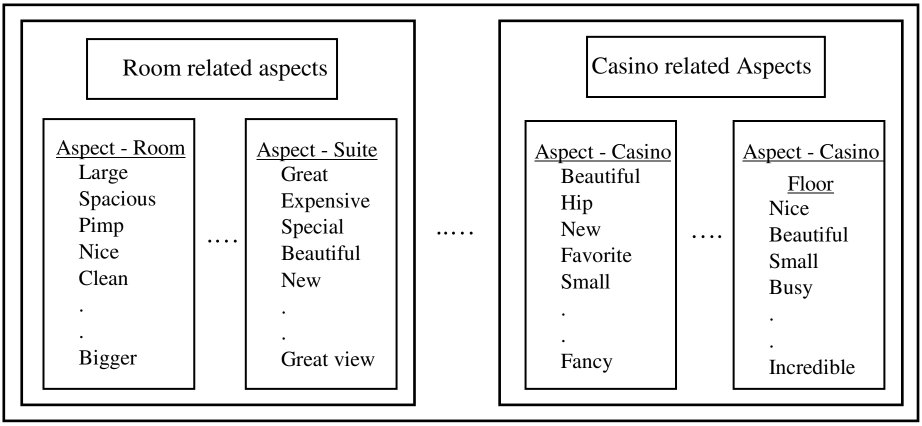

Fig. 2. Example of summary generated for aspects

An example of summary generated for aspects by our method is shown in figure 2. Different aspects in a group along with the descriptor for each aspects form a summary for the reviews which is easy to understand. One advantage of our method is that the aspect-descriptor pairs extracted from reviews can be grouped with any algorithm to produce summaries for aspects and their descriptors.

5.3 Results

Evaluation of Aspect-Descriptor Pair Discovery

The results for aspect-descriptor extraction obtained from manually labelling first 750 sentences in the dataset are presented in Table 4. It is notable that in our results, along with explicit Aspect-Descriptor pairs, our system could identify most descriptors that were used for expressing opinions about the business entity as the aspect like for "Stylish and modern" gives pairs ("forbusiness", "stylish") and ("forbusiness", "modern").

Table 4. Performance metrics

Precision	**Recall**	**F-score**
0.871	0.894	0.882

Evaluation of Aspect Clustering

For evaluation of our clustering method, we used the reviews of a particular business from the collection, from which we selected first 500 reviews, which after segmentation consisted of 6784 review sentences. We compared Simset and MCL algorithms with a modified version of LDA [12]. It takes as input a set of documents, and outputs groups of terms, and each group belongs to a topic. For input to this version of LDA (denoted as ***DLDA***), all words except the aspects in the documents are removed and only distributional information of aspects is used for grouping of aspects. The topic modeling parameters were set to their default values. The number of iteration of Gibbs sampling is set to 2000. Number of topics is set to 20.

MCL algorithm takes only inflation parameter I as input. Low value of inflation parameter produces coarser clustering, and high value fine-grained clustering. In the experiments, we adjusted I for each experiment which gives the best performance. We experimented with MCL by giving different features and their combinations as input to the algorithm. The different settings are- Only value of co-occurrence based similarity set as edge weights in graph (experiment denoted by ***MCL-pmi***), only value of Wordnet based similarity set as edge weights in graph (experiment denoted by ***MCL-word***), only value of distributional similarity of descriptors) set as edge weights in graph (experiment denoted by ***MCL-desc***) and maximum value of a features set as edge weights in graph after normalization (experiment denoted by ***MCL-mix***). In MCL-word and MCL-desc, only those pairs of aspects are considered for calculation which have a minimum co-occurrence support in the dataset, to avoid grouping unrelated aspects together.

Only needed parameters in Simset are the different threshold values we have used which we set as 7.0 for feature 1 (co-occurrence based similarity), 0.75 for feature 2 (WordNet based similarity) and 0.75 for feature 3 (descriptor based similarity) after a series of manual evaluations.

In this section we will show some objective results of our clustering. Measures like perplexity and topic coherence are often used for evaluation for clustering, however perplexity does not reflect the semantic coherence of aspects and can sometimes be contrary to human judgments and topic coherence cannot take into account the knowledge base feature used by us as it relies upon only co-occurrence statistics. So for evaluation of our system, we have used Purity and Entropy as metrics as in [23]. Since we did not have any form of ground truth for the dataset, we had to evaluate the performance of clustering manually. Purity P_j of each cluster j and the total purity of clustering solution are defined as follows

$$P_j = \frac{1}{n_j} Max\left(n_j^i\right)$$

$$Purity_{total} = \sum_{j=1}^{m} \frac{n_j}{n} P_j$$

where P_j is the purity and n_j is size of the jth cluster. n is the sum of sizes of all clusters.

Entropy measures the purity of clusters class labels. The smaller the entropy values, better the clustering is. The entropy and the total entropy are defined as follows.

$$E(S_r) = -\sum_{i=0}^{q} \frac{n_r^i}{n_r} log \frac{n_r^i}{n_r}$$

$$Entropy_{total} = \sum_{j=1}^{m} \frac{n_j}{n} E(S_r)$$

where q is the number of classes, and n_r^i is the number of documents of the ith class that were assigned to the rth cluster. The entropy of the whole clustering is defined as the sum of the individual cluster entropies weighted according to the cluster size.

The results for evaluation of clustering algorithms are summarized in Table 5. We did our evaluation of top 20 clusters according to the number of aspects contained by them for every method. It clearly show that the algorithms MCL-mix and Simset outperform all other baseline methods by a large margin.

Table 5. Experimental results for clustering algorithms

Clustering algorithm	Average no. of aspects in a cluster	Total Purity	Total Entropy
LDA	20.0	0.29	3.24
MCL-word (I=4.0)	15.1	0.65	1.50
MCL-desc (I=2.0)	8.8	0.57	1.68
MCL-pmi (I=5.0)	17.9	0.70	1.32
MCL-mix (I=5.0)	13.6	**0.71**	**1.16**
Simset	18.0	**0.75**	**0.84**

6 Discussion

The following paragraphs discuss the performance and analyze the errors observed, in increasing order of performance.

DLDA performs worst of all methods. Since DLDA depends heavily on distributional information and only considers unigrams. Aspects such as “design” and “decor” are very less likely to be put in the same groups. The results of MCL-desc have been confounded by generic descriptors for aspects, like “good” and “amazing”, which are often used for a lot of aspects (e.g., food), alongside specific descriptors such as “hot”, “tasty” and so on. If only common descriptors were used for aspects, then even unrelated aspects like “bar”, “tub”, etc. got high similarity scores resulting in poor results.

MCL-word was able to give put aspects like in “design” and “decor” in similar groups. However, if aspects were unavailable in Wordnet, it was not able to group similar aspects together. Aspect terms which were not related in the dataset, but were related in meaning and in Wordnet, got high Wordnet similarity scores and will be grouped together. Although we have set a condition for minimum support required for consideration of Wordnet similarity, still aspects like “staff” and “crowd” which although occur with minimum support, but yet are unrelated get good similarity scores in Wordnet. MCL-pmi performed slightly better.

MCL-mix considered both context and Wordnet relation into account. It proved that combining multiple criteria for aspect similarity results in better clustering. Finally, Simset clustering gives best result overall with 5% increase in purity and 40 % lower entropy than MCL-mix. The major difference between both was that in MCL, edges were given weights and in Simset, all edges had same weights but they were removed if similarity values were less than a threshold. The increase in performance can be attributed to removal of noise due to removal of edges. For example, in MCL, aspects with moderate similarity values like “game” and “pool party” were considered in clustering and if there were not enough connections of these aspects with other aspects, they would be clustered together, but in Simset such aspects with moderate value similarities are not considered for clustering.

7 Conclusion and Future Work

In this paper, we study the problem of aspect discovery and clustering. We first discovered aspect-descriptor pairs from reviews. Then we proposed three features and metrics for aspect similarity and an unsupervised clustering method. The aspect similarity features proposed performed worked well with both clustering algorithms and have proven to be better than baseline method. The experiments are preliminary, and our method has yet to be tested on different datasets and domains. More parts of speech like verbs can be considered as descriptor words. Sometimes review about an aspect may extend to more than one sentence of a review, for such possibilities, a co-reference method is needed. In future work, we plan to generate natural language summaries of aspect clusters, to highlight the constituent aspects and their descriptors in a meaningful manner.

References

1. Popescu, A.M., Etzioni, O.: Extracting product features and opinions from reviews. In: Proceedings of Human Language Technology Conference/Conference on Empirical Methods in Natural Language Processing, HLT-EMNLP 2005, Vancouver, CA, pp. 339–346 (2005)
2. Miller, G.A.: WordNet: A Lexical Database for English Comm. ACM 38(11), 39–41 (1995)
3. Qiu, G., Liu, B., Bu, J., Chen, C.: Opinion word expansion and target extraction through double propagation. Computational Linguistics 37, 9–27 (2011)
4. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proceedings of ACM-KDD, pp.168–177 (2004)
5. Yi, J., Niblack, W.: Sentiment mining in WebFountain. In: Proceedings of the 21st International Conference on Data Engineering, ICDE 2005, pp. 1073–1083. IEEE Computer Society, Washington, DC (2005)
6. Wu, Y., Zhang, Q., Huang, X., Wu, L.: Phrase dependency parsing for opinion mining. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, vol. 3, pp. 1533–1541 (2009)
7. Liu, B., Hu, M., Cheng, J.: Opinion Observer: Analyzing and Comparing Opinions on the Web. In: Proceedings of WWW, pp. 342–351 (2005)
8. Carenini, G., Ng, R., Zwart, E.: Extracting knowledge from evaluative text. In: Proceedings of International Conference on Knowledge Capture, pp. 11–18 (2005)
9. Guo, H., Zhu, H., Guo, Z., Zhang, X.X., Su, Z.: Product Feature Categorization with Multilevel Latent Semantic Association. In: Proceedings of International Conference on Information and Knowledge Management, pp.1087–1096 (2009)

10. Matsuo, Y., Sakaki, T., Uchiyama, K., Ishizuka, M.: Graph based word clustering using a web search engine. In: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, EMNLP 2006, pp. 542–550. Association for Computational Linguistics, Stroudsburg (2006)
11. Blei, D.M., Ng, A.Y., Michael, J.: Latent dirichlet allocation. Journal of Machine Learning Research 3, 993–1022 (2003)
12. Titov, I., McDonald, R.: Modeling online reviews with multi-grain topic models. In: Proceedings of the 17th International Conference on World Wide Web, pp. 111–120 (2008)
13. Titov, I., McDonald, R.: A joint model of text and aspect ratings for sentiment summarization. In: Proceedings of ACL, HLT, pp. 308–316 (2008)
14. Kim, S., Zhang, J., Chen, Z., Oh, A., Liu, S.: A Hierarchical Aspect-Sentiment Model for Online Reviews. In: Proceedings of AAAI, pp. 526–533 (2013)
15. Liu, K., Xu, L., Zhao, J.: Syntactic Patterns versus Word Alignment: Extracting Opinion Targets from Online Reviews. In: Proceedings of ACL 2013, pp. 1754–1763 (2013)
16. Mukherjee, A., Liu, B.: Aspect Extraction through Semi-Supervised Modeling. In: Proceedings of ACL, pp. 339–348 (2012)
17. Liu, Q., Gao, Z., Liu, B., Zhang, Y.: A Logic Programming Approach to Aspect Extraction in Opinion Mining. In: Proceedings of the 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), WI-IAT 2013, vol. 01, pp. 276–283. IEEE Computer Society, Washington, DC (2013)
18. Hoang, H.H., Kim, S.N., Kan, M.Y.: A re-examination of lexical association measures. In: Proceedings of Workshop on Multiword Expressions (ACL-IJCNLP), pp. 31–39 (2009)
19. Li, Y.H., Bandar, Z., McLean, D.: An Approach for Measuring Semantic Similarity Using Multiple Information Sources. IEEE Trans. Knowledge and Data Eng. 15(4), 871–882 (2003)
20. Jones, K.S.: A statistical interpretation of term specificity and its application in retrieval. Journal of Documentation 28(1), 11–21 (1972)
21. Rada, M., Courtney, C., Carlo, S.: Corpus-based and knowledge-based measures of text semantic similarity. In: Proceedings of the 21st National Conference on Artificial Intelligence, vol. 1, pp. 775–780 (2006)
22. Dongen, S.V.: A cluster algorithm for graphs, Ph.D. dissertation, University of Utrecht (2000)
23. Zhai, Z., Liu, B., Xu, H., Jia, P.: Clustering product features for opinion mining. In: Proceedings of the 4th ACM International Conference on Web Search and Data Mining, pp. 347–354 (2011)