JIS Accepted for Publication
By the Journal of Information Science: http://jis.sagepub.co.uk

Article

Discovering Aspects of Online Consumer Reviews

Journal of Information Science
1–15

DOI: 10.1177/0165551510000000
jis.sagepub.com
SAGE

Kaheer Suleman
University of Waterloo, Canada

Olga Vechtomova
University of Waterloo, Canada

Abstract
In this paper we propose a fully unsupervised approach for product aspect discovery in on-line consumer reviews. We apply a two-step hierarchical clustering process in which we first cluster words representing aspects based on the semantic similarity of their contexts and then on the similarity of the hypernyms of the cluster members. Our approach also includes a method for assigning class labels to each of the clusters. We evaluated our methods on large datasets of restaurant and camera reviews and found that the two-step clustering process performed better than a single-step clustering process at identifying aspects and words refering to aspects. Finally, we compare our method to a state-of-the-art topic modelling approach by Titov and McDonald, and demonstrate better results on both datasets.

Keywords
Opinion Mining; Clustering; Aspect Extraction

1. Introduction

The success of on-line review websites such as Zagat[1], TripAdvisor[2] and Yelp[3] has led to the presence of large numbers of on-line consumer reviews. These reviews have become an important factor to consumers for deciding to make a purchase. Therefore, it is important for service providers, such as those mentioned above, to be able to present the information in the reviews in an easy to understand format. One such method that has been successful is to present a summary based on the rateable aspects associated with the review domain. Aspects are defined as the properties of an object that can be commented on by a reviewer [1][2]. Applications that attempt to process the information contained in consumer reviews often rely on the availability of domain-specific semantic lexicons, however, the hand creation of such lexicons can be both time consuming and expensive. Furthermore, the semantic classes of aspects contained within the reviews may not be known ahead of time. For example, although one could easily come up with a number of common aspects for hotels, it would be difficult to determine a list of aspects associated with electric shavers. This difficulty arises from the fact that product aspects may have very little to do with the function of the product itself and may be subjective. For example, the portability of an electric shaver has little to do with shaving but it could be an important aspect for consumers.

In this paper we propose a method for automatically discovering semantic classes from on-line consumer reviews related to product aspects. The proposed methods are based on word clustering using the well known distributional similarity method proposed by Lin in [3]. While Lin's similarity method was used for the clustering of semantically similar words in [3, 4], it was never evaluated on the task of discovering aspects in consumer reviews. The novel contributions of our work are summarized below.

We believe that this is the first work to use the clustering method based on Lin's similarity for the discovery of product or service aspects in consumer reviews. We developed a novel method for the second-stage clustering based on hypernyms of the cluster members, returned in the first clustering stage. The benefit of the two-stage clustering is that it leads to more coarse-grained clusters, compared to a single-stage clustering. To our knowledge this is the first work to propose two-stage clustering using distributional similarity as the first stage and hypernyms as the second. Finally, we

Corresponding author:
Kaheer Suleman, University of Waterloo, University of Waterloo, 200 University Ave. W, Waterloo, Ontario, Canada, N2L 3G1
k.suleman@uwaterloo.ca

propose a method for automatically labeling clusters using the hypernyms of the cluster members. We believe that we are the first to do this for the task of product or service aspect discovery in consumer reviews.

The rest of this paper has the following structure: in Section 2 we review previous works that are similar to ours, in Section 3 we describe our proposed methodology, in Section 4 we present the results of our experiments and in Section 5 we discuss future work and conclude.

2. Related Work

The work presented in this paper can be placed into two major areas: semantic class discovery and aspect extraction. The goal of aspect extraction is to identify features or aspects related to products in consumer reviews. Methods for product/feature extraction usually fit into one of two groups: supervised and semi-supervised/unsupervised. Supervised methods require annotated corpora, often for training of statistical classifiers. Sequential classification approaches that use models such as lexicalized hidden markov models (HMM) [5] and conditional random fields (CRF) [6] treat the problem similarly to named entity recognition [7]. In both of these approaches, the aspects are labeled according to their position, for example, in the sentence "the pasta sauce was great", "pasta" would have the label B-FOOD to indicate that it is at the start of a phrase and "sauce" would have the label "I-FOOD" to indicate that it is inside a phrase. Other, non sequential supervised models, include maximum entropy [8] and support vector machines [9].

The major drawback of supervised techniques is that the annotated training corpora are difficult and costly to obtain. This has led to an increased focus on semi and unsupervised methods. Hu and Liu [10] and Popescu and Etzioni [11] propose methods for aspect extraction based on the assumption that product features/aspects are likely to occur as noun phrases. In both of these works, high frequency noun phrases are extracted as candidates and then filtered using a scoring function. Raju et al. [12] apply group average agglomerative clustering based on the number of ngrams in common between the words. Topic modeling approaches such as [1, 13] treat each aspect as fine-grained distributions over the words in the reviews. Traditional topic modelling techniques often have difficulty since document level co-occurrence is often not enough to detect aspects [1]. In order to address this, Titov and McDonald [1] propose MG-LDA, in which two topic distributions are determined: local and global. Global topics are assumed to be fixed accross a single document whereas local topics change based on the context of words. Documents are represented as a collection of sliding windows of size *n*. Each window has an associated local topic distribution and a preference distribution which represents the preference for local versus global topics. Words can be sampled from any window as long as the sentence containing the word is contained within the window.

The second area related to our work is that of semantic class discovery. Lin [3] proposes a method for grouping words based on the similarity of the contexts the words appear in. Lin defines a context to be the set of typed dependencies in which a word participates. Lin and Pantel [14] build upon the work of Lin [3] by applying a two-step clustering process. In the first step, the data set is separated into subsets. A maximal clique algorithm is then used to find cliques for each term. They define a clique to be a set of words such that each word belongs to the top *n* similar words for every other word in the clique. In the second step, clique centroids are computed by averaging the feature vectors of the words in the cliques returned from the first step. Finally, the cliques are merged together based on the similarity between their centroids. Other clustering methods include [15, 4].

One of the major shortcomings of unsupervised methods such as clustering is that they do not provide any mechanism for returning class labels. A number of methods that address this have been proposed [15, 16, 17]. Pantel and Ravichandran [17] derive a label based on the words that appear in dependency relationships with the cluster members. In [15], the class labels are derived from hypernyms that appear with conjuctions involving the word “other”. They used patterns that are based on two patterns for discovering hypernyms proposed by Marti Hearst [18], for example “NP_X and other NP_Y” and “NP_X or other NP_Y”. [18]. Our method differs greatly from that of [15] by the types of patterns used to gather hypernyms. They focus only on hypernyms that appear in conjunctions, whereas we derive hypernyms using all of Hearst’s patterns [18]. Furthermore, we believe that the methods in [15, 17] would not be as effective for reviews since class labels often do not appear in reviews. For example, in the phrase "The pizza was delicious", there is no mention of the class label “food" for “pizza". Cimiano and Staab [16] address this problem by using a web search engine to retrieve hypernyms. Although both our method and that of [16] use the web to gather hypernyms, they differ greatly in how they are used. In [16] the hypernyms are used to guide a single round of clustering in order to build a taxonomy similar to that of WordNet [19]. We use hypernyms in a two-step clustering approach in order to create more coarse-grained clusters and assign class labels to them.

3. Methodology

Lin [3] suggests that words that appear in similar contexts have similar meanings. This relationship is even more evident when it comes to product features/aspects and their subjective modifiers. For example people often use different words when referring to “food" than when they are referring to the “waitstaff". It would be very common for someone to say the wait staff was “polite", but, it is very unlikely that someone would refer to their meal as “polite". This suggests that by grouping words together based on the similarity of their contexts one could build lexicons of semantically related review aspects. To this end, we propose the following method:

(1) Extract candidate nouns from the corpus (Section 3.2);
(2) Cluster each candidate where the cluster similarity is the average Lin’s similarity between the words in the clusters (Section 3.3);
(3) Extract hypernyms from the web for each noun candidate (Section 3.4.1);
(4) Merge clusters based on the hypernyms that are associated with the cluster (Section 3.4.3);
(5) Label each cluster according to hypernym representation (Section 3.4.4).

3.1. Typed Dependencies

Before starting our process, we applied the Stanford dependency parser [20] in order to extract dependency relations between the words in the reviews. Dependency relations are defined as asymmetric binary relations between a head word and its dependent [21]. In a typed dependency parse [3][20], labels representing the grammatical relationships, such as determiner (det), nominal subject (nsubj) and copula (cop), are assigned to the relations (see Example 1).

Example 1 Sample Dependency Triples for “The pizza was good”

det(pizza-2, The-1)
nsubj(good-4, pizza-2)
cop(good-4, was-3)

3.2. Selection of Candidate Words

The first step in our method is to extract a list of candidates for clustering from the corpus. Wu et. al [22] found that 98% of the product features in their corpus appeared as single nouns. Based on this finding, we begin by first extracting all single nouns as our initial candidate set. In order to reduce the number of errors caused by the part-of-speech tagger, we filter all words that appear as nouns less than 50% of the time as well as those that only appear once. Due to the subjective nature of the reviews, we believe that aspect words are those that have opinions expressed about them, therefore, after extracting the initial set of candidates we further filter nouns that do not appear with at least one modifier. We define a modifier as a word whose part of speech is one of JJ, JJR or JJS (adjectives and their comparative and superlative forms) and is a dependent in one of the following dependencies [20]: nsubj (nominal subject), amod (adjectival modifier), rcmod (relative clause modifier).

3.3. Clustering

3.3.1. Word Similarity

After generating the list of candidates, we compute a similarity matrix S such that, for words T_i and T_k, $S_{ik} = Sim(T_i, T_k)$. We compute similarity between two words, based on the amount of contextual information shared by the two words versus the amount of information in the description of the two words. The description of a word is defined as the total set of dependency relations in which the word participates [3]. The equation for this, as presented in [3], is written as follows:
Let *w*,*w*' be words and *r* be the dependency relationship between them.

$$I(w, r, w') = \log \frac{f(w, r, w') \times f(*, r, *)}{f(w, r, *) \times f(*, r, w')} \quad (1)$$

We use (*) to denote all tuples that match the relationship pattern defined by the other words. For example, *f*(*good*,*amod*,*) would refer to the total number of tuples where *good* has the relationship *amod* (adjectival modifier) with any word.
Let *T(w)* be the set of tuples *(w',r)*, where *r* is a dependency relationship, such that *I(w,r,w')*>0

$$Sim\ (w_1, w_2) = \frac{\sum_{(r,w)\in T(w_1)\cap T(w_2)} I(w_1,r,w)+I(w,r,w_2)}{\sum_{(r,w)\in T(w_1)} I(w_1,r,w) + \sum_{(r,w)\in T(w_2)} I(w,r,w_2)} \tag{2}$$

In the above equation, *f*(*w*,*r*,*w*') represents the frequency with which words *w* and *w*' participate in the relationship *r*.

3.3.2. Hierarchical Clustering

Once we have computed the similarity matrix, for all the candidate words, we apply hierarchical agglomerative clustering. We begin the process with each word belonging to a single cluster containing only itself. After each iteration, we select two clusters for merging based on the following cluster similarity measure:
Let C_1 and C_2 be disjoint clusters

$$Sim(C_1, C_2) = \frac{1}{\|C_1\|\|C_2\|} \sum_{c\in C_1} \sum_{d\in C_2} Sim(c, d) \tag{3}$$

We continue this process until the maximum cluster similarity score is lower than the stopping threshold α. Our method for selecting α is described in Section 4.2.1.

3.4. Hypernym Clustering

The clustering method proposed in the above section leads to a large number of small clusters representing very fine-grained semantic classes, however, what is often required for applications are higher level classes. An example of this can be found on restaurant review websites where aspect categories such as “food", “environment" and "value" are given ratings. We propose a method for further clustering based on the hypernyms of the members of each cluster. Hypernym is a word, the meaning of which includes the meanings of more specialized words (hyponyms). For example, the word “animal” is the hypernym of the word “cat” since every cat is an animal. Since the meaning of a hypernym is conceptually broader than the meaning of its hyponyms, we believe that by clustering based on the hypernyms of each cluster member we can obtain more coarse-grained clusters representing higher level classes.

3.4.1. Hypernym Discovery

The first step in the process is to gather a set of hypernyms for each member of the clusters. Our method for hypernym discovery is similar to the method proposed in [23]. We begin by generating a set of extraction patterns, based on the six patterns presented in [18], for each of the clustered words. Since the method in [23] was used to extract hyponyms, we modify the patterns to extract hypernyms. The patterns are then used as a query to a commercial search engine.

Many of the clustered words have multiple meanings depending on the context of their use, for example, the word “chair” could be the short form for “chairman" or a piece of furniture. In order to prevent this from reducing the quality of the search engine results, we append the word representing the domain ("restaurant" in the case of restaurant reviews) to each of the search queries. We believe that this does not change the level of supervision since the domain would often be known ahead of time or could be discovered using an automatic method.

After removing HTML tags from the documents, we process each of the documents using a shallow parser [24] in order to identify noun phrases. Since our goal is to generate coarse-grained clusters representing general categories we chose to only extract the head words for the hypernyms. For each of the six patterns (e.g. “NP_X and other NP_Y”) and for each word in our clusters (e.g. “fries”), we generate a regular expression pattern (e.g., “fries and other NP”), which is then used to extract the head word of the noun phrase (NP) making up the hypernym. We chose to use the last word of the noun phrase as the head word. For example, the regular expression pattern “fries and other NP” would match a phrase “fries and other fast foods”, from which our method extracts “foods” as the hypernym, since “foods” is the head word of the noun phrase “fast foods”. Choosing the last word as the head of the noun phrase is sufficient because for noun phrases that do not have prepositional phrases or conjunctions the last word is almost always the head. Since hypernyms are by definition more general they are unlikely to contain any of these types of phrases.

3.4.2. Hypernym Representations

Once we have gathered the set of hypernyms for each word in our clusters, we compute the hypernym representations for each cluster. We define the hypernym representation of each cluster as the unique set of hypernyms generated by each word cluster member. Each word in the hypernym representation is given a score using the following equation:
Let c be a cluster and let h be a hypernym in the hypernym representation of c, Let N be the total number of clusters and let N_h be the total number of clusters containing h in its hypernym representation

$$Score_c(h) = TF_c(h) * IDF(h) \tag{4}$$

$$TF_c(h) = \#\ of\ terms\ in\ cluster\ c\ with\ h\ as\ a\ hypernym$$

$$IDF(h) = \frac{N}{N_h}$$

In order to reduce the potential for erroneous hypernym extractions caused by errors in both the clustering and the hypernym extraction phase, we remove hypernyms that only occur with a single member in the cluster.

3.4.3. Clustering of Hypernym Represenations

The clusters are then selected for merging according to the similarity of the hypernym representations. We follow a similar procedure to our method presented in Section 3.3.2; however, since we do not have dependency information regarding the hypernyms, we cannot use Lin's similarity. Instead, we treat each cluster as a vector where the *i*th index represents the score for the *i*th hypernym. We calculate the similarity between two clusters as follows:

$$F(w) = WEIGHT(w) \tag{5}$$

$$CosineSim(C_1, C_2) = \frac{\sum_i C_{1_i} C_{2_i}}{\sqrt{\sum_i (C_{1_i}^2)}\sqrt{\sum_i (C_{2_i}^2)}} \tag{6}$$

$$Overlap(C_1, C_2) = \frac{\sum_{w \in c_1 \cap c_2} F(w)}{\min\ (\sum_{w \in C_1} F(w), \sum_{w \in C_2} F(w))} \tag{7}$$

Equations 6 and 7 result in a local score and therefore allow clusters containing only common, low importance words to be merged. To address this problem we compute the global maximum weighted cluster overlap and compare the maximum score at each round to it using the following equations:
Let $C_1$and C_2 be candidates selected for merging and let T be the intersection of $C_1$and C_2. Let K be the total set of clusters.

$$WCO(C_1, C_2) = \sum_{w \in T} \min\left(F_{C_1}(w), F_{C_2}(w)\right) \tag{8}$$

$$\frac{WCO(C_1, C_2)}{\max_{k_i, k_{j, i \neq j \in K}} WCO(k_i, k_j)} > threshold \tag{9}$$

In the above equation, $F_{c_i}(w)$ is the weight for word *w* given cluster C_i according to Equation 4.

3.4.4. Automatic Labeling of Clusters

One of the major shortcomings of fully unsupervised techniques such as clustering is that they do not provide a way for labeling the clusters. In order to overcome this, we propose a method for automatic labeling of clusters based on hypernyms of cluster members that are identified using the method described in the previous section.

We begin by ranking the hypernyms for each cluster according to the score computed using Equation 3. We then take the top *K* hypernyms as the label for the cluster. Examples of sample clusters and their labels are shown in Table 1.

Table 1. Clusters and their Top 5 Ranked Hypernyms

Cluster Words	Top 5 Hypernyms
Carmel [sic], caramel, nut, pumpkin, banana, apple, coconut, mango, lemon, buttermilk, truffle, rum	Flavors, drinks, desserts, dessert, dishes
Fireplace, fire, oven, wood, coal, charcoal	Sources, elements, tools, fuels, risks
Pancakes, muffins, cookies, breads, meatballs, pastries, cakes, pies, biscuits, slices, fruits, pieces, veggies, cravings, oysters	Foods, food, fare, items, dishes

4. Evaluation

4.1. Experiment Setup

We evaluated our methods using two restaurant datasets presented in [25] and one camera dataset [10]. The first restaurant dataset, henceforth referred to as large restaurant dataset, consisted of 157,865 English restaurant reviews taken from a major commercial review database representing 38,782 restaurants located in North America. The reviews were pre-processed to remove html tags and encodings [26]. In order to create an evaluation set (clustering gold set) we had four annotators, the two authors and two graduate students that were not involved in the research, label each of the cluster candidates (after filtering described in Section 3.2). Each annotator was given the task of assigning one of the following labels to each candidate in the list:

(1) Food
(2) Service: staff
(3) Service: attributes
(4) Physical environment (e.g., furniture, interior/exterior, location)
(5) Ambiance (e.g., music, lighting)
(6) Clientele
(7) Menu selection (e.g., variety, wines list)
(8) Value/price

The class labels were selected based on their appearance as categories on various restaurant review websites. The candidates for which at least three of the annotators agreed on the class were taken as the final evaluation set. We found that three out of four annotators agreed on the class label for 69.1% of the candidates. The final set contained 600 words. A large number of words that had a class label belonged to the category food. Since we were using a sample of restaurant reviews from a commercial review website, we feel that this imbalance would be seen in any real world restaurant review corpus. The full distribution can be seen in Figure 1.

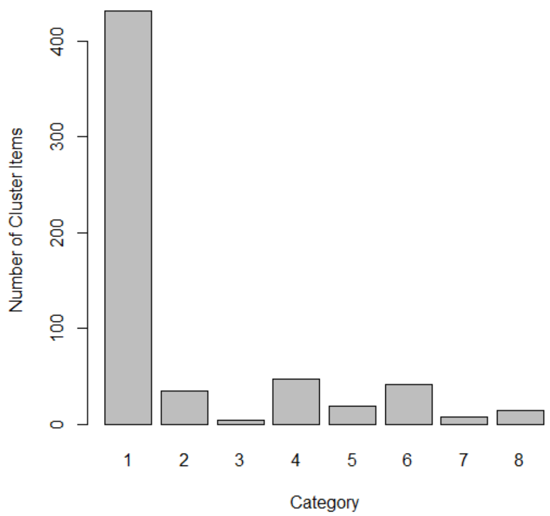

Figure 1. Distribution of Gold Set Words for Large Restaurant Dataset

For our second restaurant dataset (small restaurant dataset) we used the annotated set presented in [25]. The set contained a random selection of 600 reviews taken from the same corpus of restaurant reviews. This set was only used for evaluating the cluster labeling since it was too small to evaluate the performance of the clustering methods. Two annotators manually assigned labels for the phrases in the reviews. Each annotator labeled a non-overlapping set of 300 reviews and a third annotator went through each of the annotations and made corrections. In order to measure inter-annotator agreement, a set of 50 reviews were annotated with dish names by both annotators. The inter-annotator agreement was measured using the *agr* metric (Equation 10) [25, 27]. We found that our annotators had an overall *agr* score of 0.7.
Let *A* be the set of strings labeled by annotator *a* as class *C* and let *B* be the set of strings labeled by anotator *b* as class *C*

$$agr(a|b) = \frac{\|A\ matching\ B\|}{\|A\|} \tag{10}$$

In order to calculate the final *agr* score we calculate both *agr*(*a*|*b*) and *agr*(*b*|*a*) and we take the average. We calculate *agr*(*b*|*a*) using Equation 10 with *b* and *a* swapped.

Our camera test set was derived from a set of Amazon reviews [28] relating to photography products. Each review in the larger Amazon corpus has a product group number and a manufacturer id. In order to obtain our subset of camera reviews, we first selected all reviews with a product group number corresponding to *photography*. We then selected all reviews, from this set, with a manufacturer id equal to either “Nikon”, “Kodak” or “Canon”. We chose these specific ids since they represent major camera manufacturers. After this process, the total number of reviews in our final set was 5289. While this dataset is much smaller than our large restaurant dataset, the size is similar to other works that have used product reviews in evaluation [1, 29].

The evaluation set was set up in a similar manner to the first test set using the following categories which were verified by a domain expert:

(1) Physical characteristics (e.g dimensions, weight, ergonomics, build quality)
(2) Price
(3) Lens (e.g zoom, angle, apperture size)
(4) Image quality (e.g resolution, pixel count, flash, video)
(5) Storage
(6) Battery, power
(7) Software (e.g user interface, features/functions of software)

(8) Screen, display (display resolution, display type)
(9) Viewfinder
(10) Connectivity, wiring
(11) General characteristics

Some of the words had ambiguous meanings, for example, the word “size” could belong to both the general characteristics category or the physical characteristics category. In order to account for this we allowed each annotator to assign multiple labels to each of the candidates. Each label on which at least three of the annotators agreed was taken to be a gold label, for example, if annotator A, B and C all agreed that the word “lens” should have labels 3 and 4, then “lens” would be assigned both 3 and 4 as labels. Despite allowing for multiple labels to be assigned we found that after removing words on which annotators did not agree, there were only 4% of words with more than one label. Similar to the restaurant dataset, 69.4% of the words were kept in the final gold set.

We found that, unlike the large restaurant dataset, the camera dataset did not have a majority of words belonging to a single class. The full distribution of gold set words is shown in Figure 2.

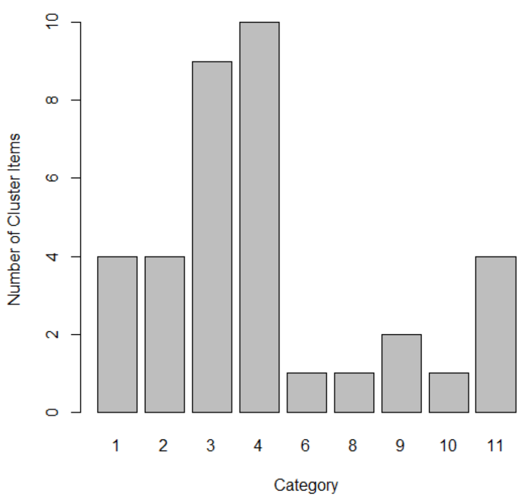

Figure 2. Distribution of Gold Set Words for Digital Camera Dataset

4.2. Results

4.2.1. Hierarchical Clustering

For the purpose of evaluation we had two baselines: no clustering (all singleton clusters, where each cluster contains one word only) and complete clustering (all clusters put in one group). We then evaluated our clustering method with stopping thresholds at intervals of 0.02. The intervals were in the range starting at 0.02 and ending at 0.10. Due to a large number of candidates not belonging to any of the categories in the gold set, we expected that our clustering would result in a large number of singleton classes; therefore, we evaluated our methods with and without removing singleton clusters.

In order to compare the performance of each of the clusterings, we calculated both micro averaged (Equation 8) and macro averaged (Equation 9) F_1 score. F-measures are a popular method for evaluating hierarchical clustering [29,30]. Micro averaged F-measures give higher weights to dominant classes and therefore can be biased towards the dominant class in imbalanced datasets. On the other hand macro averaged F-measures treat each class equally and therefore can artificially increase the importance of smaller goldset clusters. In order to show that our method performs similarly on both balanced and unbalanced datasets we chose to measure both macro and micro averaged F-measures. The results of our experiment are summarized in Tables 2, 3, 4 and 5.
Let C be a clustering, G be a gold clustering and N be the total number of cluster candidates.

$$\text{Let } P(c_i) = \frac{|c_i|}{N} \text{ and } P(c_i, g_i) = \frac{|c_i \cap g_i|}{N}$$

$$F_\beta(c_i, g_i) = (1+\beta^2)\frac{Precision(c_i,g_i) \times Recall(c_i,g_i)}{\beta^2 Precision(c_i,g_i) + Recall(c_i,g_i)}$$

$$Precision(c_i, g_i) = \frac{P(c_i,g_i)}{P(c_i)}$$

$$Recall(c_i, g_i) = \frac{P(c_i,g_i)}{P(g_i)}$$

$$F_{micro}(C,G) = \sum_{g_i \in G} \frac{|g_i|}{N} \max_{c_i \in C} F_\beta(c_i, g_i) \quad (11)$$

$$F_{macro}(C,G) = \frac{1}{|G|} \sum_{g_i \in G} \max_{c_i \in C} F_\beta(c_i, g_i) \quad (12)$$

Tables 2 to 5 show that there is a considerable increase in the F_1 score from applying our clustering technique for both the restaurant and camera review sets.

Table 2. Micro Averaged F_1scores for Single Stage Clustering on Large Restaurant Dataset

Stopping Threshold	Singletons	0.1	0.08	0.06	0.04	0.02	Single Cluster
Cluster F_1	0.025	0.122	0.159	0.153	0.207	0.330	0.244
# of Clusters	3903	1903	1764	1556	1209	943	1

Table 3. Micro Averged F_1scores for Single Stage Clustering on Camera Dataset

Stopping Threshold	Singletons	0.1	0.08	0.06	0.04	0.02	Single Cluster
Cluster F_1	0.393	0.473	0.565	0.608	0.663	0.488	0.136
# of Clusters	121	108	100	93	75	50	1

Table 4. Macro Averaged F_1scores for Single Stage Clustering on Large Restaurant Dataset

Stopping Threshold	Singletons	0.1	0.08	0.06	0.04	0.02	Single Cluster
Cluster F_1	0.125	0.247	0.279	0.258	0.282	0.260	0.116
# of Clusters	3903	1903	1764	1556	1209	943	1

Table 5. Macro Averaged F_1scores for Single Stage Clustering on Camera Dataset

Stopping Threshold	Singletons	0.1	0.08	0.06	0.04	0.02	Single Cluster
Cluster F_1	0.42	0.457	0.482	0.500	0.510	0.41	0.08
# of Clusters	121	108	100	93	75	50	1

For our large restaurant dataset, the large score from the second baseline (single cluster) for the micro averaged F-measure was likely caused by a large percentage of "food" related words relative to the other aspects. Out of the 600 aspect related words that our annotators agreed upon, 431 were related to the food aspect. We define aspect related words as all words that our annotators assigned to a class. The large class imbalance also explains the improved performance when comparing micro averaged F-measure as opposed to macro averaged F-measure. Although, the macro averaged F-measure scores were lower than the micro-averaged F-measure, our methods still show an improvement over our baseline.

We also observed that our clustering methods resulted in a large number of clusters that were small in size. For a stopping threshold of 0.02, 60% of the resulting clusters had fewer than three members and 20% of them consisted of only a single member. This was likely caused by a large number of candidates not belonging to any class. On average, our annotators found that 60% of the candidates fell into this category. We also found that our methods had difficulty with dish words that were also used to describe types of restaurants. These include words such as sushi, pizza, salad, etc. These words were clustered into the same clusters which contained words related to the physical properties of a restaurant such as place, bar and restaurant.

The results for the camera dataset were similar to those of the restaurant reviews. We found that the optimal stopping value was close to that of the restaurant reviews. This suggests that a non corpus specific value could be selected as the stopping value and still produce near optimal results.

4.2.2. Hypernym Clustering

We evaluated the three methods presented in Section 3.4.3 for merging clusters using hypernym representations. For evaluation, we used the same annotated sets used in the previous experiment. For each method we evaluated the merging thresholds at intervals of 0.1 lying on a 2-dimensional grid with the axes representing the thresholds. For both thresholds the intervals were in the range of 0 to 1. Tables 6 and 7 summarize the micro averaged F_1 score for the optimal merging thresholds at different stopping thresholds from the first stage of clustering.

Table 6. Micro Averaged F_1scores for Two-Stage Clustering on Large Restaurant Dataset

Similarity Method				Stopping Threshold	
	0.1	0.08	0.06	0.04	0.02
Cosine Similarity	0.245	0.278	0.301	0.447	0.429
Weighted Cluster Overlap (IDF Weights)	0.120	0.172	0.237	0.297	0.429
Cosine Similarity (IDF Weights)	0.137	0.162	0.240	0.378	0.429

Table 7. Micro Averaged F_1scores for Two-Stage Clustering on Camera Dataset

Similarity Method				Stopping Threshold	
	0.1	0.08	0.06	0.04	0.02
Cosine Similarity	0.546	0.618	0.638	0.663	0.488
Weighted Cluster Overlap (IDF Weights)	0.523	0.565	0.612	0.667	0.488
Cosine Similarity (IDF Weights)	0.508	0.618	0.623	0.663	0.488

The results of our experiment clearly demonstrate an improvement in cluster F_1 after clustering based on the hypernyms on the large restaurant dataset. The micro averaged F_1 score at a stopping threshold of 0.02 was 0.429 as compared to 0.330 for single stage clustering. The overall maximum micro averaged F_1 score (0.447) was found at 0.04 using cosine similarity and no IDF weights. This was different from the single stage clustering which had its maximum micro averaged F_1 score (0.330) at a stopping threshold of 0.02.

The performance of all three methods in our dataset was identical for the stopping threshold of 0.02. This was likely due to the fact that the initial clustering phase resulted in a smaller number of larger clusters. At this stopping threshold, the application of each of the methods resulted in the merging of these large clusters. The overall F_1 score was then dominated by the F_1 scores of the large clusters.

The performance improvement on the camera dataset was not as pronounced as the restaurant dataset. There was only a slight improvement (0.004) over the maximum micro averaged F_1 score (0.667) after the two-stage clustering compared to the single-stage clustering (0.663). A possible reason for this is that many of the categories for the digital camera are not “single noun” categories such as food but rather abstract concepts such as “general characteristics”. Use of the head word would then result in a larger number of generic hypernyms such as “characteristics” and “factors” to appear across all hypernym representations. One possible way to improve on this is to cluster based on increasing “generality”. In this case we consider the single stage clustering to be the most specific clustering and the head word clustering to be the most general. As an intermediate step we can perform a round of clustering based on the full noun phrase instead of just the head word. While the full phrase hypernyms are more general than the individual words used in the single round, they are able to account for more descriptive concepts such as ”optical characteristics”. We leave an evaluation of this to future experiments.

4.2.3. Automatic Labeling

We evaluated our method for labeling clusters based on two datasets: one from the items labeled as food from our large restaurant dataset and the other from the words annotated as "food" from the small restaurant dataset (see section 4.1 for a detailed description of these datasets). Since the purpose of the experiment was to evaluate how well our methods performed at assigning labels, all words that did not exist in our clusters were removed from the small restaurant set. Since our annotators only assigned a single label to each of the words in our evaluation sets we were not able to directly compare the class labels of each word to the assigned label of its cluster due to possible word mismatch. For example, our annotators assigned the category "food" to the words "tenderloin", "swordfish" and "venison" while our cluster label contained "dishes" and "meats". Since "dishes" is a synonym for "foods" it is a valid label for the cluster, however, if we only used our annotations we would have considered it incorrect. To address the word mismatch problem in our evaluation, we used a list of 12 seed words related to the category "food" (see Table 8 for the complete list of seed words). The seed set was built by selecting high scoring terms that appeared in the total set of hypernyms for the clusters, (i.e the potential set of all labels) that contained the word "food" as an inherited hypernym in WordNet [19] as well as the words "food" and "foods". We then took all clusters, for which at least one of the *K* label words was in our seed set. We compared all words in these clusters, which we consider as "food words", to the manually annotated food words in our evaluation set. In order to calculate precision on the small restaurant dataset we filtered all words that did not appear at least once in any of the 600 reviews. The results of our experiment are summarized in Tables 9 and 10.

Table 8. List of seed words for food related words

Food, dishes, ingredients, drinks, vegetables, salads, seafood, sides, meats, fruits, appetizers

Table 9. Precision and Recall for Automatic Labeling of Food Words on Small Restaurant Dataset for various *K* values

K	Precision	Recall
1	0.638	0.653
2	0.618	0.708
3	0.597	0.731
4	0.588	0.744
5	0.591	0.812
6	0.570	0.826
7	0.579	0.863

Table 10. Precision and Recall for Automatic Labeling of Food Words. *K*=5

Annotation Set	Precision	Recall
Large Restaurant Dataset	0.541	0.835
Small Restaurant Dataset	0.591	0.813

Table 9 shows that our method performs well at assigning class labels to clusters. Even when we assign a label based on a single hypernym (K = 1) we are able to recall 65% of the words. Using the top 5 hypernyms (ranked according to Equation 3) we are able to recall 80% of the words while still maintaining a precision close to 60%. Table 10 shows that our method scales to the larger dataset as well. The precision and recall values for our large dataset were similar to the smaller restaurant dataset. As described in section 4.2.2, our clusters contained only the head words of the phrases and therefore resulted in a large number of very general hypernyms e.g "characteristic" or "properties". While these hypernyms suffice as possible cluster labels they do not match well the manually assigned class labels in our camera dataset, therefore we leave the evaluation of the automatic labeling on the camera data set for future experiments.

4.3. Comparison to MG-LDA

Recently, topic modelling has become a popular approach for modelling consumer reviews. We compared our clustering technique to a method based on multi-grain latent dirichlet allocation (MG-LDA), proposed in [1], on both our large restaurant dataset and our camera dataset. In order to compare the two methods, we performed a similar set of preprocessing steps.

Titov and McDonald [1] state that their method requires that the number of global topics is at least double the number of local topics. For the restaurant reviews we chose to use the same value (50 global, 20 local) as the optimal values described in [1] for restaurant reviews. They claim that the quality of the local topics is not affected by the number of global topics as long as the number of global topics meets the requirement stated above. Furthermore, they state that one can keep creating local topics up to the optimal after which the topics no longer represent aspects. Since Titov and McDonald did not evaluate using a digital camera corpus we computed the optimal number of local topics for the dataset by performing a linear scan at increments of size 1 starting from 1 to 15. The results for the optimal number of local topics (11) are shown in Tables 13 and 14. Since the original code for MG-LDA was not publicly available, for the purpose of our evaluation we used an open source implementation of MG-LDA, written by Masanao Ochi[4].

In order to compare our methods against MG-LDA, we had to address two problems. The first problem was that MG-LDA results in a distribution over the words for each topic. When comparing soft clustering against our hard clustering, the soft clustering has the advantage that words could be counted towards the precision and recall in multiple clusters. In order to address this, we computed a hard clustering version of MG-LDA. The hard clustering was computed by assigning a word to the topic that had the highest probability of generating that term. For example if there were two topics T_1 and T_2 and $P(restaurant|T_1) = 0.5$ and $P(restaurant|T_2) = 0.3$ then restaurant would be assigned to cluster T_1. This can be written, more formally, using the following equation:

Let T be the set of Topics and let w be a word

$$Cluster(w) = argmax_{t \in T} P(w|t) \tag{13}$$

The second problem is that every word in the corpus is included in the clustering. Since our evaluation set only contained cluster candidates, our method would have had an advantage in terms of overall cluster precision. In order to address this we only considered the top k words in each topic distribution as a cluster and we only considered cluster candidate words as described in Section 3.2. All other words were ignored. Similarly to our previous experiments we compute both micro and macro averaged F_1 scores (see Equations 11 and 12). Tables 11, 12, 13 and 14 show the results of our experiment at optimal cut-off/stopping threshold.

Table 11. Large Restaurant Dataset Micro Averaged F_1 scores for MG-LDA vs Hierarchical Clustering at Optimal Cut-off/Stopping Thresholds

Method	Score
Single Stage Clustering (0.02 stopping threshold)	0.330
Two-Stage Clustering (0.02 stopping threshold)	0.429
MGLDA Soft Clustering (304 cut-off)	0.378
MGLDA Hard Clustering (97 cut-off)	0.264

Table 12. Large Restaurant Dataset Macro Averaged F_1scores for MG-LDA vs Hierarchical Clustering at Optimal Cut-off/Stopping Thresholds

Method	Score
Single Stage Clustering (0.04 stopping threshold)	0.282
Two-Stage Clustering (0.04 stopping threshold)	0.317
MGLDA Soft Clustering (33 cut-off)	0.162
MGLDA Hard Clustering (97 cut-off)	0.151

Table 13. Camera Micro Averaged F_1scores for MG-LDA vs Hierarchical Clustering at Optimal Cut-off/Stopping Thresholds

Method	Score
Single Stage Clustering (0.04 stopping threshold)	0.663
Two-Stage Clustering (0.04 stopping threshold)	0.667
MGLDA Soft Clustering (6 cut-off)	0.508
MGLDA Hard Clustering (10 cut-off)	0.355

Table 14. Camera Macro Averaged F_1 scores for MG-LDA vs Hierarchical Clustering at Optimal Cut-off/Stopping Thresholds

Method	Score
Single Stage Clustering (0.04 stopping threshold)	0.510
Two-Stage Clustering (0.04 stopping threshold)	0.511
MGLDA Soft Clustering (6 cut-off)	0.291
MGLDA Hard Clustering (1 cut-off)	0.261

Table 11 shows that after hypernym clustering our method has a higher micro averaged F_1 score compared with both the soft clustering and hard clustering versions of MG-LDA on the restaurant dataset. On the other hand, the soft clustering version of MG-LDA did outperform the single stage clustering. When comparing the methods based on macro averaged F_1, our methods outperformed both the soft and hard clustering versions of MG-LDA. This suggests that our method performs better at identifying clusters that are not the majority class. It should be noted that Titov and McDonald [1] also found that MG-LDA did not perform as well on the restaurant reviews. They suggested that this was likely caused by the fact that restaurant reviews are small in size, only 3 or 4 sentences on average. Furthermore, the reviews in our corpus were written for a variety of restaurant types, such as Chinese, Japanese and Italian. Titov and McDonald [1] suggested that the performance of MG-LDA might be better on restaurant reviews belonging to a specific type of restaurant.

Overall MG-LDA did perform much better on digital camera reviews as compared to the restaurant reviews, however, both the soft and hard clustering versions performed worse than our method. Unlike the restaurant reviews, the performance of MG-LDA was lower than that of the single-stage clustering. The macro and micro results for the digital camera reviews show a similar trend to the restaurant reviews with our methods outperforming MG-LDA under both measures.

5. Conclusions and Future Work

We introduced a two-step hierarchical clustering process based on semantic similarity and hypernym similarity and demonstrated that it greatly increases the cluster F_1score over a single-step clustering process. Furthermore, we demonstrated that Lin's similarity [3] can be an effective similarity measure for clustering words found in on-line consumer reviews. Finally, we showed that we can overcome the inability of clustering methods to assign labels using a simple technique based on the hypernyms of the cluster members.

Our experiments demonstrate that clustering can be an effective method for discovering aspects in an unsupervised manner. Despite this, our methods still ended up returning a large number of clusters. In future work, we will work on reducing this. Due to the characteristics of the datasets used in our experiments, it was not possible to perform a statistical significance test. The only way to perform a statistical significance test would have been to split the datasets into smaller subsets and apply the clustering methods to each subset. This, however, is not possible for two reasons: firstly, the resulting subsets would be too small, and secondly, one of our datasets has a large number of words belonging to a single class (Figure 1). If we split it into smaller subsets we would likely not have enough words of the other classes in each subset. We leave it for future experiments to evaluate the method on more datasets and perform a significance test.

One of the major applications of the methods presented in this paper is that of aspect based sentiment. User opinions on various product aspects found in reviews can be different from the overall sentiment of the review. An example of this is in a restaurant review, where a negative review may contain positive comments on the food, but also negative comments on both the physical environment and the service. This suggests that it is important to measure sentiment with respect to the individual aspects found within a review. Most current systems require a predefined set of categories for which reviewers are asked to assign ratings. Since our methods do not rely on the knowledge of the domain or the aspects present in the reviews, they identify aspect categories automatically, and allow ratings to be assigned without having to manually predefine categories. In future experiments, we will evaluate how our methods can be used to automatically identify and assign ratings to unknown aspects.

Finally, in order to fully evaluate the effectiveness of our system, one would need to evaluate its performance in a "live" environment. More specifically, the overall goal of such a system is to provide categories that are helpful to consumers of on-line reviews as they perform a specific task, such as selecting a restaurant. Future work should involve a user study using a live implementation in which the "helpfulness" of the categories returned is evaluated.

Notes

1. http://www.zagat.com/
2. http://www.tripadvisor.com/
3. http://www.yelp.com
4. https://github.com/m-ochi/mglda/blob/master/

Funding

This research was supported by the Natural Sciences and Engineering Research Council of Canada [grant number RGPIN 261439-2013].

References

[1] Ivan Titov and Ryan McDonald. Modeling online reviews with multi-grain topic models. In *Proceedings of the 17th International Conference on World Wide Web*, pages 111–120, New York, NY, USA, 2008. ACM.

[2] Benjamin Snyder and Regina Barzilay. Multiple aspect ranking using the good grief algorithm. In *In Proceedings of the Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistic*, pages 300–307, 2007.

[3] Dekang Lin. Automatic retrieval and clustering of similar words. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 2*, pages 768–774, Stroudsburg, PA, USA, 1998. Association for Computational Linguistics.

[4] Patrick Pantel and Dekang Lin. Discovering word senses from text. In *Proceedings of the eighth ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, pages 613–619, New York, NY, USA, 2002. ACM.

[5] Wei Jin and Hung Hay Ho. A novel lexicalized hmm-based learning framework for web opinion mining. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 465–472, New York, NY, USA, 2009. ACM.

[6] Fangtao Li, Chao Han, Minlie Huang, Xiaoyan Zhu, Ying-Ju Xia, Shu Zhang, and Hao Yu. Structure-aware review mining and summarization. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 653–661, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[7] Andrew Eliot Borthwick. *A maximum entropy approach to named entity recognition*. PhD thesis, New York, NY, USA, 1999. AAI9945252.

[8] Sasha Blair-goldensohn, Tyler Neylon, Kerry Hannan, George A. Reis, Ryan Mcdonald, and Jeff Reynar. Building a sentiment summarizer for local service reviews. In *Proceedings of WWW-2008 workshop on NLP in the Information Explosion Era*. 2008.

[9] Jianxing Yu, Zheng-Jun Zha, Meng Wang, and Tat-Seng Chua. Aspect ranking: identifying important product aspects from online consumer reviews. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, pages 1496–1505, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[10] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 168–177, New York, NY, USA, 2004. ACM.

[11] Ana-Maria Popescu and Oren Etzioni. Extracting product features and opinions from reviews. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 339–346, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[12] Santosh Raju, Prasad Pingali, and Vasudeva Varma. An unsupervised approach to product attribute extraction. In *Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval*, pages 796–800, Berlin, Heidelberg, 2009. Springer-Verlag.

[13] Samuel Brody and Noemie Elhadad. An unsupervised aspect-sentiment model for online reviews. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistic*, pages 804–812, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[14] Dekang Lin and Patrick Pantel. Induction of semantic classes from natural language text. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 317–322, New York, NY, USA, 2001. ACM.

[15] Sharon A. Caraballo. Automatic construction of a hypernym-labeled noun hierarchy from text. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 120–126, Stroudsburg, PA, USA, 1999. Association for Computational Linguistics.

[16] Philipp Cimiano and Steffen Staab. Learning concept hierarchies from text with a guided agglomerative clustering algorithm. In *Proceedings of the Workshop on Learning and Extending Lexical Ontologies with Machine Learning Methods*, pages 6–15. 2005.

[17] Patrick Pantel and Deepak Ravichandra. Automatically labeling semantic classes. In *Proceedings of HLT/NAACL 2004*, 2004.

[18] Marti A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th Conference on Computational Linguistics - Volume 2*, pages 539–545, Stroudsburg, PA, USA, 1992. Association for Computational Linguistics.

[19] George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine Miller. Wordnet: An on-line lexical database. *International Journal of Lexicography*, 3:235–244, 1990.

[20] Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. Generating typed dependency parses from phrase structure trees. In the *5th edition of the International Conference on Language Resources and Evaluation*, 2006.

[21] Joakim Nivre. Dependency grammar and dependency parsing. Technical Report MSI 05133, Växjö University, School of Mathematics and Systems Engineering, 2005.

[22] Yuanbin Wu, Qi Zhang, Xuanjing Huang, and Lide Wu. Phrase dependency parsing for opinion mining. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3 - Volume 3*, pages 1533–1541, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[23] Olga Vechtomova and Stephen E. Robertson. A domain-independent approach to finding related entities. *Information Processing and Management*, 48(4):654–670, 2012.

[24] Vasin Punyakanok and Dan Roth. The use of classifiers in sequential inference. In *Neural Information Processing Systems*, pages 995–1001. MIT Press, 2001.

[25] Olga Vechtomova. A method for automatic extraction of multiword units representing business aspects from user reviews. *Journal of the American Society for Information Science and Technology*. 65(7):1463–1477, 2014.

[26] Mohamad H Ahmadi. A semi supervised approach to the construction of semantic lexicons. PhD thesis, University of Waterloo, 2012.

[27] Janyce Wiebe, Theresa Wilson, and Claire Cardie. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 39(2-3):165–210, 2005.

[28] Nitin Jindal and Bing Liu. Review spam detection. In *Proceedings of the 16th international conference on World Wide Web*, pages 1189–1190, 2007.

[29] Jianxing Yu, Zheng-Jun Zha, MengWang, Tat-Seng Chua,. Aspect ranking: identifying important product aspects from online consumer reviews

[30] Michael Steinbach, George Karypis, and Vipin Kumar. A comparison of document clustering techniques. *In Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining Workshop on Text Mining*, 2000.

[31] Florian Beil, Martin Ester, and Xiaowei Xu. Frequent term-based text clustering. In *Proceedings of the 8th International Conference on Knowledge Discovery and Data mining*, pages 436–442, New York, NY, USA, 2002. ACM.