A Two-Iteration Clustering Method to Reveal Unique and Hidden Characteristics of Items Based on Text Reviews

Alon Dayan
Department of Computer Science
University of Haifa, Israel
dayanalon@gmail.com

Osnat Mokryn
Department of Information Systems
University of Haifa, Israel
omokryn@is.haifa.ac.il

Tsvi Kuflik
Department of Information Systems
University of Haifa, Israel
tsvikak@is.haifa.ac.il

ABSTRACT
This paper presents a new method for extracting unique features of items based on their textual reviews. The method is built of two similar iterations of applying a weighting scheme and then clustering the resultant set of vectors. In the first iteration, restaurants of similar food genres are grouped together into clusters. The second iteration reduces the importance of common terms in each such cluster, and highlights those that are unique to each specific restaurant. Clustering the restaurants again, now according to their unique features, reveals very interesting connections between the restaurants.

Categories and Subject Descriptors
H.3.3 [INFORMATION SEARCH AND RETRIEVAL]: Information filtering, Clustering;

I.2.7 [NATURAL LANGUAGE PROCESSING]: Text analysis;

I.5.4 [Applications]: Text processing

General Terms
Algorithms; Experimentation

Keywords
Clustering; Textual Reviews; Text Mining; Latent Connections

1. INTRODUCTION
People nowadays share online their opinions about a wide variety of products and services, such as restaurants, hotels and movies in the form of reviews. Online reviews often contain rich and valuable information about these products and services. Eliciting relevant information from reviews is challenging due to the plethora and diversity of reviews written by different people referring to different aspects of the products. Yet, it is a well-known and widely researched problem [1]–[4] with commercial applications [5].

An item may have many features. A Restaurant, for example, has its name, location, opening hours, food genre, special dishes, ambience, etc. Some of these features are explicitly noted as part of the structured data of the reviews site (e.g. the restaurant's name, opening hours). Other features are embedded in the reviews and require text analysis tools to be pulled out from the text. Popular dishes of a restaurant (such as Pasta and Pizza in Italian restaurants or Beer in Pubs), for example, could easily be extracted using common Term Weighting Schemes [6].

The main focus of this paper is a third group of unique features, which are not part of the structured information in the reviews site, and cannot be revealed easily by traditional methods. Special attractions of restaurants, the attitude of the owner towards costumers, and the common audience of a restaurant are examples of such features. The method we present extracts such features automatically from reviews in an unsupervised manner. Moreover, we use the extracted features to reveal latent connections between items (in this work we use restaurants as our items). Examples vary. Restaurants of different genres but with the same chef; Recommended restaurants for a post cultural event pastime; Restaurants liked by specific audiences (i.e., students, sports fans) or recommended for a desired ambiance (i.e., romantic venues, lunch venues); Restaurants with a common unique culinary specialty; And even restaurants with a similar rare attraction of projecting movies on the walls while serving dinner.

The method is built of two similar iterations of terms extraction, then applying a weighting scheme for weighting them and then clustering the resultant set of vectors. The first iteration is common practice in text clustering [7], [8]. The weighting scheme highlights meaningful terms in the representative vectors of restaurants, by analyzing each restaurant with respect to the whole corpus. Then the clustering algorithm groups these vectors to sets of restaurants with similar meaningful terms. Not surprisingly, the obtained clusters contain restaurants of similar food-genres (Italian restaurants in one, Pubs in another and so on).

The second iteration is the main novelty of this work. A weighting scheme is now applied again, this time by analyzing each restaurant with respect only to other restaurants in its food-genre. In this way, it reduces the importance of common terms in the genre, and highlights those that are unique to each restaurant. Clustering the restaurants again, now according to their unique features, reveals interesting connections between the restaurants.

For example, the cluster in Figure 1 contains four restaurants, which at first glance don't seem to have any special connection: two Italian restaurants, one Bar, and one American and B&B place. However, these restaurants share a very unique attraction – they all project movies on the walls while serving dinner. This unique feature is revealed only in the second iteration of clustering, as in the first iteration these restaurants were clustered to different groups, according to their food-genres.

Even more interesting is the fact that different weighting schemes reveal different latent connections. When another weighting scheme is used (different from the one used in the above example), we get a cluster with only two of the four restaurants. This time, the latent connection between them has nothing to do with movies. It turns out that the two restaurants are adjacent to each other, share the same audience, and even have a joint toilet room. The meaning of this example is that a set of restaurants may have more than one latent connection between them, and that different weighting schemes may discover these different connections.

WWW 2015 Companion, May 18–22, 2015, Florence, Italy.
ACM 978-1-4503-3473-0/15/05.
http://dx.doi.org/10.1145/2740908.2741707

Figure 1 - Two different types of latent connections

All experiments in the paper are tested on a large corpus of text reviews of restaurants. However, it is important to emphasize that the method is general, and could be applied on different types of items as well. Moreover, the method does not use any external information besides the textual reviews, and the whole process is done without human intervention and in a completely unsupervised manner.

2. Related Work

Online textual reviews contain meaningful information for users, manufacturers, and service providers. In recent years, an intensive research effort is directed towards investigating the potential of using the content of reviews [2], [4], [9]. Obviously, the huge amount of reviews requires automatic algorithms for their analysis. Term Weighting Schemes [6], Clustering [10], and Aspect Identification [9] are important approaches in representing and analyzing texts automatically. Before describing our method, we first briefly review these areas.

2.1 Term Weighting Schemes

The Bag of Words (BOW) [6] is a common and highly popular model for document representation. A weighted vector of terms represents each document, where the term's weight indicates the importance of the term in representing the document. The methods that assign weights to terms are called Term Weighting Schemes. TF-IDF [6], which is based on the frequency of terms in a document (TF), and in the entire corpus (IDF), is one of the most popular ones. The main intuition of the scheme is that a term represents a document well if it is frequent enough in it, and at the same time infrequent in the whole corpus.

Notable extensions to TF-IDF are Okapi BM25 [6] and LTU [11] that consider in their formulas supplemental parameters (e.g., the lengths of documents in the corpus), in addition to the TF and IDF values. CBT [12] (Cluster Based Term weighting scheme) is a cluster-specific version of TF-IDF. In addition to the frequency of terms in documents, CBT also considers the frequency of terms in clusters. The highest weight is assigned when a term is very frequent in one cluster and uncommon in other clusters. In such a way, documents are not considered only as separate entities, and the contextual relations between them (represented by the clusters) are also considered in the process of weighting the terms.

Another approach is based on the Kullback-Leibler (KL) divergence [13]. KL is a known statistical information measure that measures the total difference between two probability distributions P and Q. Unlike TF-IDF and other term weighting schemes, KL is not intended specifically for text analysis. KLD [14] and LIT [7] are two weighting schemes that make the required adaptations in order to use KL for the purpose of term weighting. The input distributions P and Q refer to the distribution of words in documents and the distribution of words in specific categories. TF-ICF [8] is another scheme, which unlike most schemes, does not require any knowledge on the distribution of term frequencies in the entire collection of documents. Instead, it uses a sufficiently large and diverse static corpus as a substitute.

In our experiments, we tested our method with the TF-IDF and KL schemes. Interestingly, using different schemes enrich the overall extracted data, as they extract different unique features and reveal different latent connections between restaurants. According to this observation, more weighting schemes should be tested in future experiments.

2.2 Clustering

Clustering is an unsupervised machine learning technique with two main properties. First, it produces homogenous groups (clusters) of similar items in such a way that items within a cluster are more similar to each other than they are to items in other clusters [6]. Second, it reveals patterns within the data without the need of any additional information [15]. Unsupervised methods in general, and clustering algorithms specifically, are very popular in the domain of text analysis, and serve in a variety of grouping tasks [10]. In the context of text reviews, unsupervised methods are particularly valuable, as they are not influenced by the specific nature of reviews (short, unstructured, contain slang and misspelling).

There are many types of clustering techniques [6], [10]. In our experiments we used Affinity Propagation (AP), which is flat (clusters are not organized in any specific structure) and hard (each element is a member of exactly one cluster). The fact that AP doesn't require an external input of the number of clusters makes it more relevant for us. In the first iteration, we have no pre-knowledge about the number of main genres of the clustered items, and in the second iteration there is no way to predict the number of latent contextual connections between items.

2.3 Aspect Identification

In Aspect Identification [9], the goal is to find a set of relevant aspects of the reviewed items, such as the ambiance or service quality in a restaurant [15]. Usually, it is followed by a Sentiment Classification phase [1], [5], which aims to find the sentiment of reviewers towards the revealed aspects. Our research is mainly connected to Aspect Identification, as we also aim to extract features of items from textual reviews.

Using the frequency of terms was among the first approaches suggested for extracting aspects [9]. The idea is to look for frequent occurrences of explicit phrases in the corpus. In [16], for example, the occurrence frequency of terms is compared with their frequency in English in general. Terms that are significantly more frequent in a specific text than in general English are considered important.

Probabilistic Latent Semantic Analysis (pLSA) [17], [9] is a more complicated approach that aims to discover the main themes in documents. The core idea is that documents consist of multiple topics, which appear in different proportions in the text. Probabilistic Topic Models aim to reveal these topics and their proportions in documents, as well as the distribution of words in each topic. They assume that documents were produced using a generative process. By using tools of statistical inference, such as Gibbs Sampling, they reverse the generative process and conclude the latent variables. Latent Dirichlet Allocation (LDA) [18] is a basic and widely used such model.

The two approaches share a similar goal - both extract features in order to later find their sentiment. Using these features and sentiments, similar items can be compared in an automatic

process[5]. Our method, however, has a different goal. Instead of looking for the mutual aspects of items, we look only for features which are unique to specific items. These features can't serve as a basis for comparison between similar items, but rather enrich the description of items by emphasizing their uniqueness. In future work, it would be beneficial to also find the sentiment of these unique features.

3. THE PROPOSED METHOD

We present here a method for extracting unique features of items and reveal latent connections between them, based solely on their text reviews. The process consists of two iterations in which we extract the main features of items (restaurants in our case) and then cluster them accordingly. The first iteration is straight forward. However, in the second iteration we take a unique approach for feature selection, enabling us to reveal contextual features as well as latent connections between restaurants.

Clustering has an important role in our algorithm, and we use it in three different places. In the first iteration, it is used to group together similar items, according to their global features (terms that were assigned with high weights by the weighting scheme, such as 'Pizza' and 'Pasta' in a cluster of Italian restaurants). In the second iteration, clustering is executed in order to find items that share similar local features (these terms are meaningful for a specific item, but not for other items in its cluster. 'Movie', for example, is a very meaningful term for an Italian restaurant that projects movies on its walls, but it is not dominant at all in other members of the cluster of Italian restaurants). Between the two iterations, clustering is used again, this time in order to group genres with congruent characteristics (like Bars and Pubs) to Meta-Genres. We now elaborate on these three steps.

3.1 First Iteration – Reducing Importance Weights of Corpus-Terms

The input of our method is a collection of text reviews. In order to represent each restaurant as a weighted vector of terms, we apply a set of basic text operations. First, we merge all reviews of the same restaurant to one large document and name it a Restaurant Document (RD). Reviews are short texts, so in order to ensure sufficient information we consider only restaurants with more than 30 reviews. Then we remove stop-words, perform case folding, stemming, noun extraction, and frequency counting. At the end of this process each restaurant is represented as a vector of nouns and their frequencies [7].

We now apply TF-IDF[1] on the results, creating weighted vectors of terms, one per each restaurant. Each weight represents the importance of the term in representing the restaurant. In these vectors, the impact of terms that are frequent and common in many documents in the corpus (we name them *Corpus-Terms*) is reduced by the use of IDF.

We then cluster these representative vectors, using the affinity propagation algorithm (Figure 2 presents a schematic description of whole process). The advantage of affinity propagation is that it doesn't require a predefined number of clusters. This property fits our scenario, as we don't know in advance the number of food genres in the corpus. Theoretically, we could have taken the food genres of restaurants from the structured data of Yelp. However, in order to keep the method independent of external data, we don't use this information, but deduce it directly from the text.

[1] For the simplicity of explanation we now consider TF-IDF as our weighting scheme. However, other schemes can also be applied.

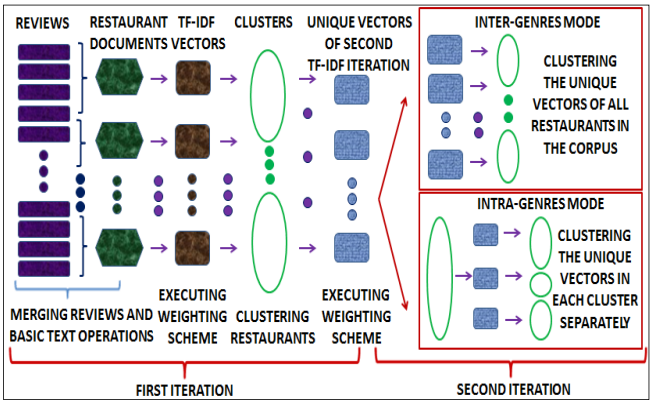

Figure 2 - A schematic representation of the algorithm

As expected, the restaurants in this stage are clustered according to their main food genres (there are 69 different genres in the corpus, such as Italian restaurants, Pubs, and so on). Accordingly, we name these clusters *Genre-Clusters*. In order to verify that the members of these clusters are indeed of the same food genre we use the concept of Meta-Genres.

3.2 Meta-Genres

After the first iteration, each obtained cluster contains restaurants of similar food-genres (*Genre-Clusters*). To verify this, we compare the resultant clusters to the genres of restaurants, as structured in the meta-data of the corpus in Yelp.

A common way to verify such a claim is to calculate the F1-measure [6], which evaluates the rate of agreement between the genres of restaurants in each cluster. A cluster with four restaurants, for example, all of them are *Pubs*, would get the highest possible F1 score due to the perfect agreement between the genres. Another cluster with four restaurants, two *Italians,* one *Japanese,* and one *Mexican*, will get a lower F1 score since the agreement is weaker.

However, in our scenario, things are a bit more complicated. A closer look at the structured genres in Yelp reveals many genres with negligible differences. For example, the genres *'Coffee & Tea'* and *'cafes'*, or the genres *'Mediterranean'*, *'Middle Eastern'*, and *'Greek'* - are mostly congruent. Restaurants of these genres serve similar dishes, and the differences between them, in terms of their food genre, are usually minor. However, technically, these are different labels, and a hypothetical cluster (C1) with three restaurants (*'Mediterranean'*, *'Middle Eastern'*, and *'Greek'*), will get a low F1 score. This is while the cluster actually does represent a group of restaurants with similar genre characteristics and should get a high score.

To resolve this issue we use *Meta-Genres* – groups of genres with similar meanings and congruent characteristics. It is important that the similarity of genres would rise from the reviews, and won't be based on human knowledge. Therefore, we use clustering again. We first characterize each structured genre by summing all representative vectors of the restaurants of this genre (for example, all vectors of Italian restaurants). Then, we cluster the obtained vectors using the affinity propagation algorithm.

In our corpus, 14 *Meta-Genres* were created. One of them, for example, contains the genres: *'Mediterranean'*, *'Middle Eastern'*, and *'Greek'* (Table 1). If we look again at the hypothetical cluster C1, its F1 score is now 1 (the highest score), because all its member restaurants share the same *Meta-Genre*. The final F1

score in our corpus, when TF-IDF is the weighting scheme, is 0.904. This score is high enough to ensure that most clusters obtained in the first clustering iteration indeed contain restaurants of similar genres.

Table 1 - Five of the fourteen obtained Meta-Genres

Meta-Genre 1	German, Hot-Dogs
Meta-Genre 2	Indian, Pakistani, Vegetarian, Caterers
Meta-Genre 3	Greek, Mediterranean, Middle-Eastern
Meta-Genre 4	Japanese, Sushi-Bars, Asian Fusion, Gluten-Free
Meta-Genre 5	Pubs, Breweries, Irish, Sports Bars

3.3 Second Iteration – Reducing Importance Weights of Genre-Terms

In the first iteration, the impact of frequent terms in the corpus was reduced by the IDF part of TF-IDF. At this point we wish to reduce the importance of terms that are common in specific food-genres (we name them *Genre-Terms)*. *'Pasta'* and *'Pizza'*, for example, are *Genre-Terms* in the Italian genre, as they are very frequent in most Italian restaurants. Mexican restaurants, however, have different *Genre-Terms*, such as *'Burrito'* and *'Tortilla'*. By reducing the weights of *Genre-Terms* we can find contextual features, which are unique for a restaurant but infrequent neither in the entire corpus nor in the restaurant's genre. In other words, we aim to reduce the weights of *'Pasta'* and *'Pizza'* merely in Italian restaurants, and of *'Burrito'* and *'Tortilla'* only in Mexican restaurants.

We do that by performing again the TF-IDF calculation. This time, however, we execute it within each of the obtained *Genre-Clusters* separately. This is a key step in our method. The separate executions ensure that IDF reduces the importance weights of only the frequent terms in each genre. As a result, we now have a vector per restaurant, in which only the unique terms that characterize the restaurant within its cluster have high weights. We name these vectors *Unique-Vectors*.

When two (or more) restaurants share similar unique terms, we say that they have a latent connection between them. We find these connections by applying clustering again, this time on the *Unique-Vectors*. As presented in Figure 2, there are two modes (inter-genres and intra-genres) for the second iteration of the method. In the inter-genres mode, the unique vectors of all restaurants are clustered again, so that restaurants of different genres could be clustered together. In the intra-genres mode, however, the second clustering iteration is executed separately in each of the clusters obtained in the first iteration. Therefore, in this mode, all the obtained clusters have the same Meta-Genre. Interestingly, the two modes produce different types of results. We explore these results in section 5.

3.4 A Representative Example

To make things more tangible, we now examine one example of the whole process thoroughly. The restaurant *'Paul K'* is described by three genre labels: *Greek*, *Mediterranean*, and *Breakfast & Brunch*. At the beginning of the process, the ten most frequent terms in the whole collection are: *Food*, *Place*, *Service*, *Restaurant*, *Time*, *Brunch*, *Duck*, *Friend*, *Dinner*, and *Menu*. Most of these terms are *Corpus Terms* – terms that are frequent in the entire corpus and common in the reviews of all restaurants (such as *Food, Place*, and *Time*). Therefore, their importance scores reduce significantly after applying TF-IDF, resulting in a new list of important terms: *Mezza*, *Riblet*, *Paul*, *Mezz*, *Spelt*, *Fryup*, *Baba*, *Ganoush*, *Halloumi*, and *Moussaka*. This list is a mixture of *Genre Terms* (common terms in the *Greek* and *Mediterranean* genres, such as *Mezza*, *Moussaka*, *Baba* and *Ganoush*) and unique terms, which are special for the specific restaurant (such as *Paul* and *Spelt*). In the following steps, our goal is to automatically highlight only the unique terms.

Interestingly, even though the restaurant has the label *Breakfast & Bruch* (*B&B*), its top terms indicate that it is more *Greek* and *Mediterranean* in its nature. Moreover, *Pancake*, *Breakfast*, and *Omelet*, which are of the most representative terms of the *B&B* genre, appear very rarely in the restaurant document (7, 4, and 1 occurrences respectively). The meaning is that the restaurant is definitely not a typical member of the *B&B* genre. For that reason the clustering algorithm positioned *'Paul K'* in the *Middle Eastern*, *Greek*, and *Mediterranean* cluster, and not in the *B&B* cluster.

We now move to the second iteration, and apply TF-IDF again in order to reduce the importance weights of *Genre-Terms*. Recall that this step is applied separately within each *Genre-Cluster* (Figure 2). The obtained top terms are now: *Paul*, *Mimosa*, *Bottomless, Brunch*, *Maple*, *Richard*, *Hayes*, *Riblet*, *Confit*, and *Duck*. These terms are obviously unique to the restaurant. *Paul* (the name of the restaurant and the owner) climbed to be the most unique term. *Richard* is the restaurant's waiter, and users tend to mention him in their reviews. *Maple Bacon*, *Confit Duck*, and *Lamb Riblets* are all popular dishes served in the restaurant. Interestingly, even though *Lamb* is the 12th most frequent term in the restaurant document (112 occurrences), it is only in the 90th position in the list of unique terms. *Maple*, however, occurs only 19 times in the restaurant document (140th in the frequencies list), but it climbed to be the 5th most unique term. This is a direct result of our mechanism. *Lamb* is a common dish in *Middle Eastern*, *Greek*, and *Mediterranean* restaurants, and therefore not unique to *'Paul K'*. *Maple*, however, is a rare ingredient in these food genres, and therefore unique to *'Paul K'*.

4. EXPERIMENTAL SETUP

Our corpus is taken from the business review site Yelp.com (Bay area, 2010). After the removal of restaurants with less than thirty reviews (to ensure sufficient information for each restaurant), our corpus contains 278,335 reviews of 877 restaurants. Figure 3 presents the distribution of reviews per restaurant. 35.8% of the restaurants have less than one hundred reviews, and 5.6% have more than a thousand. The average length of review in our corpus is 134 words. In addition to the unstructured textual part of the reviews, the data also contain a structured section with several built-in categories, including the food genres of the restaurants. There are 69 possible genre labels in the corpus. The most frequent genre is 'American (New)', and it describes 16.4% of the restaurants in the collection (144 restaurants). Nine genres

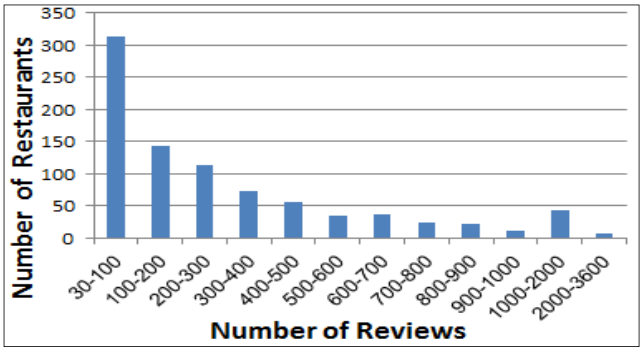

Figure 3 - The Distribution of reviews in the corpus

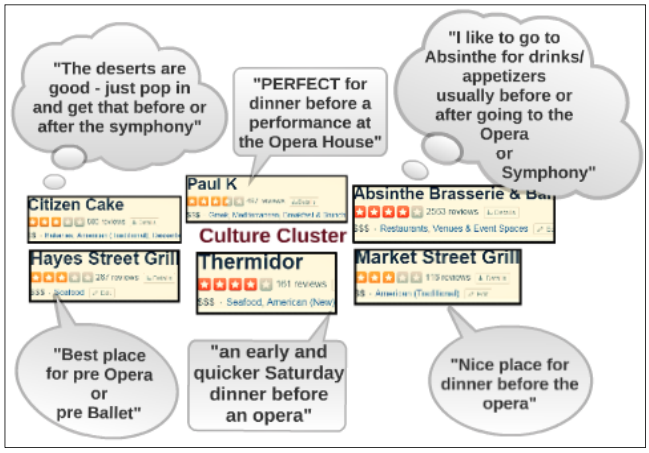

Figure 4 - A culture based cluster with six restaurants

describe four restaurants each (genres with less than four restaurants were removed to ensure sufficient information per genre). A large part of the data is multi labeled, as each restaurant has a set of labels that describe its food genres. The majority of the restaurants (46.7%) have one genre, 39.2% have two, 14% have three, and only one restaurant has four genres.

5. EXPERIMENTAL RESULTS

In this section we present the results of our method. As Figure 2 shows, the second iteration has two different modes - inter-genres and intra-genres. We dedicate a separate section to each of these modes, as they produce different types of results.

5.1 Inter-Genres Mode

After producing a unique vector of terms for each restaurant, we now come to the final step of the algorithm, where we apply the second iteration of clustering. The inter-genres mode aims to find latent connections between restaurants that are not necessarily of the same genre. This is done by analyzing the unique vector of each restaurant with respect to all other restaurants in the corpus. One example of such cluster was presented in the introduction, where restaurants of different genres that project movies on the walls were grouped together. We examine here five more interesting types of this mode. A more detailed quantitative and qualitative analysis of the results is now in progress.

The first example is of a culture-based type of clusters, where restaurants with a latent cultural connection are grouped together. One such cluster is presented in Figure 4, where six restaurants of different food genres were grouped together because people tend to visit them before or after cultural events at the opera. Notably, this is not a location-based cluster. Other restaurants in the opera surroundings exist in the corpus, but did not enter this cluster because reviewers do not relate them with cultural events. Here are some relevant reviews of restaurants in the cluster that stress this point: *"An excellent pre or post Opera dining stop", "This is a great stop after watching the ballet or theatre", "The one warning is this is the go-to place for people attending cultural events. So on any night with symphony, opera, etc., it is jammed from 6 to 7:45"*, and so on. This cluster might be used, for example, to recommend an Opera house related dining venue.

Another type of latent connection is based on the audience of restaurants. One such cluster, for example, contains five restaurants of different food genres[2]. The latent connection here is that college students tend to visit these restaurants on a regular basis. Here are several reviews that emphasize this point: "*it is a favorite among the entire student population*", "*The Korean operated Joshu-Ya have been a stable for Berkeley residents and students for many years", "This Starbucks to me is another study hall for Cal Berkeley students", "a good break for any stressed college student",* and more. Such information about the target audience of restaurants can improve recommender systems and enrich the structured data of review sites.

Another interesting type of latent connection is related to the chef or owner of the restaurants. In one such cluster, two restaurants[3] of different genres were clustered together, because both have the same chef. A reviewer of one restaurant, for example, wrote: *"Grand KUDOS to Mr. Leary for giving us beautiful and nourishing meal experiences"*. Similar voices come also from reviewers of the other restaurant*: "What genius thought this place up? Oh, I should have known... Denis Leary"*. Another cluster of the same type contains two other restaurants[4] of different food genres, as both have the same chef – Dominique Crenn.

Two additional types of latent connections that our method reveals in the Inter-Genres mode are based on location and ambiance. Location-based clusters usually gather restaurants that are located in the same street, or adjacent to familiar public spaces. One such cluster, for example, contains five restaurants[5] of different genres that are located near the Buena Yerba Gardens in San Francisco. Another cluster of this type contains five restaurants[6], which are all located in the same building. We saw in the introduction two other restaurants, which are adjacent to each other and even share the same toilet room. One example of an ambiance based cluster is of three restaurants[7]. The ambiance in these restaurants, according to reviewers, is touristic. Moreover, some reviewers even describe these places as tourist traps: "*In my judgment, this is a tourist trap meant to make you think it is nicer than it is*", "*It seems essentially like a tourist trap*", "*Leave this tourist trap to the tourists!*". Obviously, such information can be valuable for users of review sites.

5.2 Intra-Genres Mode

In the alternative intra-genres mode, the second clustering iteration is applied separately on each *Genre-Cluster* obtained in the first iteration (Figure 2). Applying the method in this manner reveals latent connections between restaurants of similar genres. Interestingly, the types of latent connections in this mode are different from those in the inter-genres mode. Comparison based clusters, for example, are cliques of restaurants of the same genre that reviewers frequently compare to each other. Naturally, people tend to compare items of the same type (two romantic restaurants for example) rather than of different types (a romantic restaurant and a hot-dog stand). Therefore, we don't expect to find comparison clusters in the inter-genres mode.

---

[2] The restaurants: *Sushi Hunter, Starbucks, Yoshu-Ya Sushi, Fondue Fred*, and *Sushi House*.

[3] *Canteen* and *The Sentinel.*

[4] *Atelier Crenn* and *Luce.*

[5] *Tropisueno, B Restaurant & Bar, Caffe Museo, Beard Papa's Cream Puffs*, and *Amber*.

[6] *Gott's Roadside, Mijita, Out the Door, The Slanted Door*, and *Miette Pattiserie*.

[7] *Caffe Museo*, *Artesa Vineyards & Winery*, and *Caffe Delucchi.*

One such comparison cluster, for example, contain three *Vietnamese* restaurants[8] (out of 28 in the corpus), which users frequently compare to each other. A reviewer of *PPQ* wrote: *"I've continued to hear about how you can get better crab and lower cost at ppq as opposed to thanh long/crustaceans but nobody mentioned how extremely poor the service was*". A reviewer of *Thanh Long* wrote: "*We consistently rotate between Thanh Long, Crustaceans, PPQ*". Another reviewer, this time of Crustacean, wrote a long comparison review about the three restaurants. Here is the part about the price: "*Lastly, the cost. Crab is expensive, cost us $46 per crab. Sounds nuts right? However, guess what, PPQ charges THE SAME $46 and so does Thanh Long!*".

In another comparison clique, three German restaurants[9] were clustered together. A reviewer of *Walzwerk* wrote: "*For German food I much prefer Suppenkuche or Speisekammer (in Alameda), but the restaurant environment at these places aren't as funky and artsy*"[10]. A reviewer of *Speisekammer* wrote: "*If you like Suppenkuchen or Walzwerk in SF, you will love Speisekammer. They have a similar selection of good dishes, but with their own original interpretation*". It is interesting to see that the relationship between restaurants in such clusters may even be stronger than just a simple comparison. See the following insight written by a reviewer of *Speisekammer*: "*Speisekammer was brought to you by the husband-wife team who were the culinary brains behind Suppenkuche"*.

Another type of latent connection is based on the specific food sub-genre (or specialty) of restaurants. Again, this type is unique for the intra-genres mode, as sub-genres can only be found among restaurants of the same genre. In one such cluster, for example, there are four Indian restaurants[11] (out of 25 in the entire corpus), which specialize in the South Indian style. Our method clearly points out this specialty by highlighting typical terms and portions, which are strongly related to this sub-genre (such as *South*, *Idli*, *Vadai*, and *Sambar*). These extracted sub genres can enrich the structured categories of restaurants in the reviews site.

There are many more types of clusters, which unfortunately can't be presented in the scope of this paper. Obviously, this kind of information can enrich the structured data of items in reviews sites as well as to sharpen the suggestions of recommender systems.

6. CONCLUSIONS AND FUTURE WORK

This work demonstrates the potential of clustering in two consecutive iterations. It reveals unique features of items, and latent connections between them, by analyzing their textual reviews. By applying the suggested method reviews sites may provide better personalized service to users when enabling them to consider unique features of items, which are usually hidden.

It is interesting to note that applying different weighting schemes result in different unique features, and reveal new latent connections between them. Hence, it would be interesting to further explore this point.

Another interesting direction would be to apply the algorithm on users instead of items, as unique characteristics of users and latent connections between them may be very useful for user profiles and recommender systems.

[8] *Crustacean Restaurant, Thanh Long, PPQ Dungeness Island*.

[9] *Speisekammer*, *Walzwerk*, and *Suppenküche*.

[10] We cite the reviews without correcting language mistakes.

[11] *Dosa on Fillmore*, *Ruchi*, Vik's Chaat, and *Dosa on Valencia*.

7. REFERENCES

[1] A. Popescu and O. Etzioni, "Extracting Product Features and Opinions from Reviews," 2004.

[2] A. Levi, O. Mokryn, C. Diot, and N. Taft, "Finding a needle in a haystack of reviews: cold start context-based hotel recommender system," in *Proceedings of the sixth ACM conference on Recommender systems*, 2012, pp. 115–122.

[3] J. McAuley and J. Leskovec, "Hidden factors and hidden topics," in *Proceedings of the 7th ACM conference on Recommender systems - RecSys '13*, 2013, pp. 165–172.

[4] C.-C. Musat, Y. Liang, and B. Faltings, "Recommendation using textual opinions," pp. 2684–2690, Aug. 2013.

[5] J. Huang, O. Etzioni, L. Zettlemoyer, K. Clark, and C. Lee, "RevMiner: An extractive interface for navigating reviews on a smartphone," in *Proceedings of the 25th annual ACM symposium on User interface software and technology - UIST '12*, 2012, p. 3.

[6] C. D. Manning and P. Raghavan, "An Introduction to Information Retrieval," 2009.

[7] W. Ke, "Information-theoretic Term Weighting Schemes for Document Clustering," in *ACM/IEEE Joint Conference on Digital Libraries*, 2013, pp. 1–10.

[8] J. W. Reed, J. Yu, T. E. Potok, B. A. Klump, M. T. Elmore, and A. R. Hurson, "TF-ICF: A new term weighting scheme for clustering dynamic data streams," in *Proceedings - 5th International Conference on Machine Learning and Applications, ICMLA 2006*, 2006, pp. 258–263.

[9] B. Liu, "Sentiment Analysis and Opinion Mining," *Synthesis Lectures on Human Language Technologies*, vol. 5, no. 1. pp. 1–167, 2012.

[10] C. Aggarwal and C. Zhai, "A survey of text clustering algorithms," *Min. Text Data*, 2012.

[11] K. Sparck Jones and P. Willett, Eds., *Readings in Information Retrieval*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1997.

[12] A. K. Murugesan and B. J. Zhang, "A New Term Weighting Scheme for Document Clustering," *7th Int. Conf. Data Min. (DMIN 2011 - WORLDCOMP 2011), Las Vegas, Nevada, USA.*, 2011.

[13] S. Kullback and R. A. Leibler, "On Information and Sufficiency," *Ann. Math. Stat.*, vol. 22, no. 1, pp. 79–86, Mar. 1951.

[14] B. Bigi, "Using Kullback-Leibler Distance for Text Categorization," in *ECIR 2003*, 2003, pp. 305–319.

[15] S. Moghaddam and M. Ester, "Opinion digger: an unsupervised opinion miner from unstructured product reviews," in *Proceedings of the 19th ACM international conference on Information and knowledge management - CIKM '10*, 2010, p. 1825.

[16] C. Scaffidi, K. Bierhoff, E. Chang, M. Felker, H. Ng, and C. Jin, "Red Opal: Product-Feature Scoring from Reviews," in *Proceedings of the 8th ACM conference on Electronic commerce - EC '07*, 2007, p. 182.

[17] D. M. Blei, "Probabilistic topic models," *Commun. ACM*, vol. 55, no. 4, p. 77, Apr. 2012.

[18] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," *J. Mach. Learn. Res.*, vol. 3, no. 4–5, pp. 993–1022, 2012.