ReEL: Review aware Explanation of Location Recommendation

Ramesh Baral, XiaoLong Zhu, S. S. Iyengar, Tao Li

School of Computing and Information Sciences

Florida International University

Miami, FL, 33199, USA

rbara012@fiu.edu, xzhu009@fiu.edu, (iyengar,taoli)@cs.fiu.edu

ABSTRACT

The Location-Based Social Networks (LBSN) (e.g., Facebook, etc.) have many attributes (e.g., ratings, reviews, etc.) that play a crucial role for the Point-of-Interest (POI) recommendations. Unlike ratings, the reviews can help users to elaborate their consumption experience in terms of relevant factors of interest (aspects). Though some of the existing systems have exploited user reviews, most of them are less transparent and non-interpretable (as they conceal the reason behind recommendation). These reasons have motivated us towards explainable and interpretable recommendation. To the best of our knowledge, only few of the researchers have exploited user reviews to incorporate the sentiment and opinions on different aspects for personalized and explainable POI recommendation.

This paper proposes a model termed as **ReEL** (Review aware Explanation of Location Recommendation) which models the review-aspect correlation by exploiting deep neural network, formulates user-aspect bipartite relation as a bipartite graph, and models the explainable recommendation by using dense subgraph extraction and ranking-based techniques. The major contributions of this paper are: (i) it models users and POIs using the aspects posted on user reviews, and it provisions incorporation of multiple contexts (e.g., categorical, spatial, etc.) in POI recommendation, (ii) it formulates preference of users' on aspects as a bipartite relation, represents it as a location-aspect bipartite graph, and models the explainable recommendation with the notion of ordered dense subgraph extraction using bipartite cores, shingles, and ranking-based techniques, and (iii) it extensively evaluates the proposed models using three real-world datasets and demonstrates an improvement of 5.8% to 29.5% on F-score metric, when compared to the relevant studies.

KEYWORDS

Explainable Recommendation; Social Networks; Information Retrieval

ACM Reference Format:

Ramesh Baral, XiaoLong Zhu, S. S. Iyengar, Tao Li. 2018. ReEL: Review aware Explanation of Location Recommendation. In *Proceedings of 26th Conference on User Modeling, Adaptation and Personalization (UMAP '18)*. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3209219.3209237>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

UMAP '18, July 8–11, 2018, Singapore

© 2018 Association for Computing Machinery.

ACM ISBN 978-1-4503-5589-6/18/07...\$15.00

<https://doi.org/10.1145/3209219.3209237>

1 INTRODUCTION

Most of the existing e-commerce systems (e.g., Amazon.com, etc.) have been facilitating users to share their consumption experience via ratings and reviews. The LBSNs have also been a useful platform to share consumption experiences on different factors of interest (e.g., price, service, accessibility, product quality, etc.). For instance, the review text "*The breakfast was awesome but the front-desk service was really bad*" implies a positive experience of the reviewer towards "breakfast" and opposite for "front-desk". The words "breakfast" and "front-desk" are known as **aspect terms** and their equivalent categories "Food" and "Service" are known as **aspects**. Such experiences from a real customer have been crucial in the purchase decision for potential customers, and product improvement for manufacturers.

Despite the usefulness, reading time and uniform interpretability of reviews have been a major concern. It would have been easier if one can summarize and explain the opinions on key aspects, for instance, (i) *place A has a good rating for food*, (ii) *place B is renowned for cleanliness*, etc. Though a dedicated community has been focusing on the extraction of such aspects and opinions [12, 45, 47], the recommendation domain can also use such aspect-based summarization to enhance and explain the generated recommendation.

The exploitation of different factors of LBSN for an efficient recommendation has been quite popular in the last decade [51, 55]. Most of the studies have focused on non-text attributes, such as categorical, temporal, spatial, and social [1–3, 48, 49] but have been less transparent and less interpretable (i.e. the factors used for recommendation are hidden from end users). Contrary to that, some of the studies [17, 18, 35, 36, 40, 41, 43, 57] have already claimed the user persuasiveness due to explainability in real-world systems. The similarity-based approaches [4, 23] have proposed user-based neighbor style (e.g., "*users with similar interest have purchased the following items...*") explanations. The item-based neighbor style (e.g., "*items similar to you viewed or purchased in the past...*"), influence style (how the users' input have influenced the generation of recommendation), and keyword-style (items that have similar description content to purchase history) can be other variants of explanations.

To the best of our knowledge, only few studies have focused on review-aware explainable recommendation. There are many factors that make this problem challenging and interesting. The aspect extraction from ambiguous and noisy text, organizing the numerous aspect terms into relevant categories (e.g., food, service, etc.), and personalization of recommendation are some of the main challenges. The aspect-based personalized explanation is challenging as it needs to handle the sentiments of each aspects, and also the individual user preferences and item features to get relevant explanation.

The ease of adaptation of arbitrary continuous and categorical attributes in a scalable manner makes the Convolutional Neural Networks (CNN) a good candidate for classification problems (e.g., [13, 25]). This also makes them ideal for a supervised review-aspect classification problem. We formulate the problem of review and aspect correlation using CNNs. This simplifies the process of mapping the user sentiments to the $(POI, aspect)$ tuples and modeling the users' aspect preferences as the aspect-POI bipartite relation. We represent such a bipartite relation using a bipartite graph, extract users' ordered aspect preferences using dense subgraph extraction and ranking-based methods, and generate an explainable POI recommendation. The core contributions of this paper are: (i) it models users and POIs using the aspects extracted from reviews and different contexts (e.g., categorical, spatial, etc.), (ii) it formulates the user preferences as an ordered aspect-POI bipartite relation, represents it as a bipartite graph, and proposes bipartite core, shingles, and ranking-based methods to generate personalized and explainable POI recommendation, and (iii) it evaluates the proposed model using three real-world datasets. As an important by-product, our model can implicitly identify the user communities and categorize them by their preferred aspects. It can also identify the implicit POI groups that are known for a set of aspects.

2 RELATED RESEARCH

The problem of aspect extraction from review text has been quite popular [8, 28, 52] for various problems (e.g., rating prediction [31], aspect-sentiment summarization [24, 34, 42], recommendation [30, 53], etc.). To the best of our knowledge, exploitation of aspects for explainable POI recommendation has been less explored. We present the relevant studies in following two categories:

Aspect-based approaches: Yang et al. [50] exploited sentiment lexicon (e.g., SentiWordNet)-based approach and defined user preferences based on tips, check-ins, and social relations but did not fully exploit user preferences at aspect level. Wang et al. [46] exploited multi-modal (i.e. text, image, etc.) topics-based POI semantic similarity but ignored aspect level preference modeling and recommendation explanation. Covington et al. [14] exploited different factors, such as users' activity history, demographics, etc., but did not incorporate opinions from user comments and also did not focus on recommendation for each aspect. Guo et al. [19] represented users, POIs, aspects, and geo-social relations with a graph and ranked the nodes to define the POI recommendations. Some of the studies [15] used the features extracted from user reviews to build user and item profiles and generated the recommendation. Zhang et al. [53] used the aspect opinions, social, and geographical attributes to generate the recommendation. Chen et al. [9] used aspect-based user preferences in their recommendation. Recently, Zheng et al. [58] adapted [13] to exploit user reviews and mapped user and item feature vectors into same space to estimate user-item rating. Our model has following advantages than [58]: (i) it uses sentiment polarity of reviews at sentence level rather than the whole review text, (ii) it learns to classify each review sentence into aspects and models users and places using these aspects and embedding of additional contexts (e.g., POI category, check-in time, etc.),

and (iii) it efficiently exploits a bipartite core extraction, shingles extraction, and ranking-based methods to extract densely connected aspects and relevant POIs for an explainable recommendation.

Explanation-based approaches: Chen et al. [11] personalized ranking based tensor factorization model and used phrase-level sentiment analysis across multiple categories. They extracted aspect-sentiment pairs from review text and used Bayesian Personalized Ranking [38] to rank the features from user reviews. Finally, feature wise preference of a user was derived using the user-item-feature cube and rank of the feature obtained earlier. Zhang et al. [56] used matrix factorization to estimate the missing values and a recommendation was made by matching the most favorite features of a user and properties of items. They used simple text templates to generate a feature-based explanation of positive and negative recommendations. However, incorporation of additional features (e.g., POI category) was not explored. Lawlor et al. [27] exploited sentiment-based approach to explain why a place might(not) be interesting to a user. For each aspect, they compared the recommended place to the alternatives and provided explanation (e.g., better (worse) than 90% (20%) of alternatives for *room quality* (*price*), etc.). However, they relied on frequency of aspects of POIs and users to get such relation and incorporation of additional features remained unexplored. He et al. [21] exploited tri-partite modeling of user-item-aspect tuples and used graph-based ranking to find the most relevant aspects of a user that match with relevant aspects of places. The common relevant aspects were used in the explanation. Li et al. [10] proposed an explanation interface to explain the tradeoff properties within a set of recommendations, in terms of their static specifications and feature sentiments. However, their interface requires users to explicitly provide their preference on different aspects.

We have found that only few of the existing studies have fused few additional attributes (e.g., social), whereas most of them had no provision for them. Most of the studies were tightly coupled to aspects and their sentiments, and analyzed influence of all aspects together. The influence of aspects among each other can have adverse impact on recommendation quality, for e.g., a place that is good in "Price" aspect might be opposite in "Service" aspect. A user who just cares about "Price" aspect might ignore some "Service" related problems in that place. So we need to minimize the influence of aspects among each other. This is crucial for aspect-based recommendation systems, and to the best of our knowledge, this direction is less explored and is still a viable research problem. We attempt to fill this gap by exploiting bipartite graph and dense subgraph extraction techniques. For a user, the most dense subgraph represents the set of most preferred aspects and places popular for those aspects. The dense subgraph extraction is followed by disconnecting the edges within the dense subgraph which ensures less interference from the aspects already discovered in previous dense subgraphs. This claim is also supported by our evaluation where one of our model **ReEL-Core** performs better than our another model **ReEL-Rank** (see Sec. 4.1, Sec. 4.3, and Sec. 5 for detail).

3 METHODOLOGY

The overview of proposed system is illustrated in Figure 1. The core components of the proposed system are as follows:

Figure 1: High level overview of system architecture

- (1) **Review preprocessing:** The review texts are splitted into individual sentences and the stop words are removed.
- (2) **Aspect term extraction:** The pre-processed review sentences are fed to the aspect extraction module to extract aspect terms. A simple two-step process is applied. First, we filter out nouns and noun phrases using some experimentally set frequency threshold. Most of the reviews focus on a set of topics, hence this approach can capture such topics [33]. Second, we use a rule-based approach [54] that adopts the dependency parsing [29] to capture the aspect terms missed in the previous step.
- (3) **Aspect-categorization:** As there can be numerous aspect terms, we narrow down them to few well-known aspects (see Table 2b) for easy computation. The aspect terms and their synsets from WordNet [16] are used to assign the best matching aspect. We select top 3 synsets to handle ambiguity of aspect terms and to capture the relevant aspect.
- (4) **Sentence-aspect training data preparation:** As the aspect extraction and labeling is not the core focus of this paper, we rely on supervised sentence-aspect classification concept. The review text (after aspect term extraction) is labeled by the aspect that has closest match to its aspect terms. The distance between aspect terms and the aspects (and their synonyms) from the WordNet [16] are used to assign the closest possible label. As we assign top 3 matching synsets, a single aspect term can have three matching aspects. The sentences with multiple aspect terms get multiple label. This labeled data is used to train the CNN-based sentence-aspect classifier. The performance of this module is defined in the evaluation section (see Section 5).
- (5) **CNN-based sentence-aspect classifier:** The review-aspect correlation module is a multi-class classifier (see Figure 1a) that classifies a review sentence into relevant aspects. Inspired from [25], we use a CNN-based classifier to label each review sentence. The network consists of a convolution, an

activation function, a max-pooling, a dense layer, and a softmax layer (see [25] for detail). The input to this classifier is word embedding of review sentences. We use Word2Vec [32] to map every word to a uniform size vector in a latent feature space. The outcome of the classifier is a bipartite relation between review and the aspects.

For every user, the classifier gives a set of sentence feature vectors (later known as *user feature vectors*) that are embedding of her preferred aspects. Similarly, for every POI, the sentence feature vectors (later known as *POI feature vectors*) are embeddings of the aspects specified in its reviews. As every user tends to mention some opinion on preferred aspects in her reviews and every place is mentioned about the aspects it was reviewed for, such vectors incorporate the aspects relevant to users and POIs. As a POI can be positively or negatively reviewed for an aspect, we extract the sentiments of each review sentence by using the trigrams around the aspect terms. The embeddings of the sentiment term [32] is concatenated to the POI feature vector. As each POI can get multiple reviews on same aspect, the POI feature vector is normalized on feature vectors of each aspect. This review-aspect bipartite relation is then used to define the POI-aspect tuples and user-aspect tuples. Such a bipartite relation can be exploited to model user preferences via ordered aspect-POI relation using bipartite graph and dense subgraphs of such graph (see Sec. 4.1, and Sec. 4.3 for details). The POI-aspect pair is supplemented with the aggregated sentiment extracted from all the review sentences.

- (6) **Recommendation generation:** This variant of proposed model is termed as Deep Aspect-based POI recommender (DAP). Besides the review text, we also incorporate additional context (e.g., categorical, spatial, etc.) into the feature vector of the POIs obtained from the classifier.

We formulate the recommendation problem as a matrix, whose rows represent a user, POI, and elements of different contexts. For each row, the check-in flag of a user to a POI is treated as the target. For instance, if a user u_i has feature vector as $\langle u_{e1}, u_{e2}, \dots, u_{em} \rangle$, a place l_j has its sentiment concatenated feature vector as $\langle l_{e1}, l_{e2}, \dots, l_{en} \rangle$, and the user u_i has visited the place l_j , then a row in the design matrix is obtained simply by concatenating the user feature vector, POI feature vector, and context vectors, and is defined as: $u_i, l_j, f_k = \langle u_{e1}, u_{e2}, \dots, u_{em}, l_{e1}, l_{e2}, \dots, l_{en}, f_{k1}, f_{k2}, \dots, f_{ko}, 1 \rangle$, where u_{ea} , l_{ea} , and f_{ka} are the a^{th} item (a real-valued number) of the feature vector of user (u_i), place (l_j), and context (f_k). The last element 1 represents the check-in flag for user-place-context tuple in the training data and represents the score to be estimated on test data.

For a user u , the context vector is concatenation of temporal, spatial, categorical, and social vectors: $\langle v_{t1}, v_{t2}, v_{t3} \rangle$, $\langle v_{dist1}, v_{dist2}, v_{dist3} \rangle$, $\langle v_{cat1}, v_{cat2}, \dots, v_{catk} \rangle$, and $\langle v_{soc} \rangle$. v_{cat1} is the multiplication of embedding vector of category cat_1 and the factor $r_{cat1} = (\sum_{l, cat=cat_1} V_u(l)) / (\sum_{l' \in u_L} V_u(l'))$ (i.e. the ratio of

total check-ins made to places with category cat_1 to that of all check-ins). $v_{dist1} = (\sum_{dist(l) \leq \epsilon_1} V_u(l)) / (\sum_{l' \in u_L} V_u(l'))$ is ratio of to-

tal check-ins on places within a threshold distance ϵ_1 (from users' home, work place or most frequently checked-in place) to that of all check-ins (we consider $\epsilon_1 \leq 1, 1 < \epsilon_2 \leq 5, \epsilon_3 > 5$ as three distance thresholds (in K.M.)). $v_{soc} = (\sum_{l \in u_{fL}} V_u(l)) / (\sum_{l' \in u_L} V_u(l'))$ is the ratio of total check-ins made on places visited due to social influence to that of all check-ins. v_{t1} is the ratio of total check-ins made in time t_1 (we use three values for time - morning, afternoon, and others (night and evening)). The POI context vector consists of category, time, and distance vectors.

A factorization machine [37] is exploited to estimate the value of the check-in flag for every user-place-context tuple. As the factorization machine has the ability to deal with additional features, a user-place pair can have multiple rows but just one row for each user-place-context tuple. So, the prediction is already personalized for the user-place-context tuple. The top-N scorers from factorization machine are further filtered out using the preferred aspects of user (determined by the frequency of aspects mentioned on her reviews) and are recommended to the users. The high-level overview of the recommendation module is illustrated in Figure 1b.

- (7) Explanation of recommendation: After getting the place-aspect bipartite relation from CNN-based classifier, we represent the user-aspect preference as a bipartite graph and generate the recommendation explanation by extracting the most dense subgraphs from this bipartite graph. We propose three different methods- a bipartite core extraction, shingles extraction, and ranking-based methods for explanation generation (see Sec. 4 for detail).

3.1 Factorization Machine

The Factorization Machine [37] formulates the prediction problem as a design matrix $X \in \mathbb{R}^{n \times p}$. The i^{th} row $\vec{x}_i \in \mathbb{R}^p$ of the design matrix defines a case with p real-valued variables. The main goal is to predict the target variable $\hat{y}(\vec{x})$ using Eqn. 1. The proposed recommendation module is formulated as a sparse matrix. The rows of the matrix are generated by concatenating the embeddings of a user feature vector, POI feature vector, and context vector. We consider the check-in flag as the target variable for each row. The proposed model is operated with the following objective function:

$$\hat{y}(\vec{x}) = w_0 + \sum_{i=1}^n w_i x_i + \sum_{i=1}^n \sum_{j=i+1}^n \langle \vec{v}_i, \vec{v}_j \rangle x_i x_j, \quad (1)$$

where w_0 is the global bias of all user-POI-context tuples, $\vec{x}$ is a concatenation of user feature vector, POI feature vector, and context vector, n is the size of input variables, $\langle \vec{v}_i, \vec{v}_j \rangle = \sum_{f=1}^k v_{i,f} \cdot v_{j,f}$, and

k is the dimensionality of factorization. The Factorization Machine can learn latent factors for all the variables, and can also allow the interactions between all pairs of variables. This makes them an ideal candidate to model complex relationships in the data.

4 EXPLANATION OF RECOMMENDATION

The POI-aspect bipartite relation derived from Sec. 3 is represented as a bipartite graph and the ordered preference of user on aspect categories is extracted and used for explanation.

4.1 Bipartite Core Extraction (ReEL-Core)

A k -core of a graph is a maximal connected subgraph whose every vertex is connected to at least k other vertices. The k -core analysis is popular for community detection, dense subgraph extraction, and in dynamic graphs. Our method for bipartite core detection is inspired from [26] where each node is assigned two scores - hub score and authority score, which are defined in terms of the outgoing and incoming edges respectively. The hub score (h_i) of a node is proportional to the sum of authority scores of the nodes it links to. The authority score (a_i) of a node is proportional to the sum of hub scores of the nodes it is linked from. Given the initial authority and hub scores of all the nodes, the scores are iteratively updated until the graph converges. For a given user, we consider all the recommended places as the seed nodes and connect them to the aspect nodes for which they have overall positive sentiments (i.e. (no. of positive opinions) > (no. of negative opinions)). This filters out the negatively reviewed places and gives us a bipartite graph as shown in Fig 2a (left graph).

We calculate the eigenvectors of the adjacency matrix of the graph to identify the primary eigenpair (largest eigenvalue). The eigenvalue is used as a measure of the density of links in the graph. The iterative algorithm gives the largest eigenvalue (primary eigenpair). The primary eigenpair corresponds to the primary bipartite core (most prevalent set of POI-aspect pairs) and non-primary eigenpairs correspond to the secondary bipartite cores (less prevalent set of POI-aspect pairs). The most dense subgraph (e.g., the right subgraph in Figure 2a with nodes AC_1, P_1, P_2 , and P_3) is extracted as the primary bipartite core. After finding the primary core, the edges relevant to this core are removed and the process is repeated

Figure 2: (a) Place Aspect Graph (AC_k = aspect k , P_i = places) (Left subgraph is a bipartite graph and the right one is a primary bipartite core), (b) Aspects, and (c) Aspect score to star ratings for a POI

on residual graph to get the next prevalent bipartite cores. Removal of edges from the primary core will still leave the nodes relevant to other aspect nodes that belong to other bipartite cores. The bipartite cores are used in the order (primary, secondary, etc.) and the aspects in bipartite cores are used to explain the recommendation.

Explanation generation: A bipartite core consists of densely connected nodes and resembles the set of place nodes which are mostly known for the relevant aspect nodes. For a user, we generate the POI-aspect relations from the ordered bipartite cores as:

Aspect 1: $POI_1, POI_2, \dots, POI_i$

Aspect 2: $POI_i, POI_j, \dots, POI_{j+k}$

.....

Aspect k : $POI_1, POI_i, \dots, POI_j$,

where each row gives the aspect from the ordered bipartite core and the relevant set of POIs that are popular for that aspect. We also generate the score of each POI_i on each aspect as:

Aspect 1: $Score_{i,1}$

Aspect 2: $Score_{i,2}$

.....

Aspect k : $Score_{i,k}$,

where $Score_{i,a}$ represents the score of POI_i by the aspect a for all users, and is defined as: $Score_{i,a} = \sum_{i=1}^k \frac{1}{i} * |core_{i,a,i}|$, where the

term $|core_{i,a,i}|$ represents the number of times the POI_i was in i^{th} bipartite core for the aspect a on all users, and k represents the ordered number of bipartite cores used (e.g., $k=1$ is for primary bipartite core, $k=2$ for secondary core, and so on). The scores computed are interpolated to the 5-star rating scheme (see Figure 2c).

As an example, the review text "*Tasty free hot breakfast and friendly staffs*", implies that the reviewer cares about the "Price" and "Service" aspects, and a primary bipartite core for this user should contain these aspects and relevant places. Given the place "Hyatt Regency" and "The Setai Miami Beach" have overall positive opinions for the "Price" aspect, they are included in the primary bipartite core (i.e. related to "Price") and the explanation is generated graphically as shown in Figure 2c and is supplemented with text as:

Recommended Place: Hyatt Regency, The Setai Miami Beach, ...;
Explanation: Popular for Price.

4.2 Dense subgraph extraction (ReEL-Dense)

This model exploits the weight of user-aspect and place-aspect relation to incorporate the extent of user preferences on aspects and the popularity information of a place through the aspects.

Figure 3a shows a basic representation of the network and extraction of dense subgraphs. The POI-aspect edge is weighted by the normalized measure of frequency of overall positive opinions on the aspect for the POI. The user-aspect edge is weighted by the normalized measure of number of times the user reviewed on the aspect. We exploit the random extraction of connected components from the network and proceed with the components having high similarity score. If γ is a random permutation applied on the homogeneous sets A and B (e.g., set A has only user nodes and set B has only aspect nodes), then their similarity score is defined as:

$$Sim^{\gamma}(A, B) = \frac{f(A, B)}{f(A) + f(B)} \quad (2)$$

where $f(A, B) = \sum_{a \in A, b \in B} W_{a,b}$, where $W_{a,b}$ is the weight of edge (a,b) that

is normalized to all the edges outgoing from node a , $f(A) = \sum_{(a,i) \in E} W_{a,i}$

is the sum of normalized weights of all edges outgoing from node a , and $f(B) = \sum_{(i,b) \in E} W_{i,b}$ is the sum of normalized weights of all

edges incidence on node b . We assume that absence of POI-aspect edge indicates that the place is not known for that aspect (e.g., the aspect is irrelevant). We can use the min-wise independent permutations [6, 7] technique to avoid exploitation on each and every permutation to find the sets with high similarity score. We use some predefined number of permutations ($c=10$) and do not focus on the min-wise independence of permutations. Algorithm 1

Algorithm 1 ShingleFinder($G = (V, E)$, c , s , k)

- 1: // G is the input graph, V is the set of vertices, and E is the set of edges, c is the number of permutations, s is the length of each set, k is the number of shingles to be extracted
- 2: initialize L as an empty list
- 3: **for** each place node **do**
- 4: **for** $j = 1$ to c **do**
- 5: get a set of s aspect nodes
- 6: find aggregated similarity for the place and aspect nodes in this set using Eqn. 2
- 7: store this set and its score in L
- 8: **return** k sets with high similarity score (*these sets are called shingles*) from L

defines shingles extraction process from a bipartite graph. For each POI, we apply Algorithm 1 to find the set of aspect nodes linked

to it and extract the k shingles for it. For each shingle, we find the list of all POI nodes that contain it. These are the POIs that are mostly reviewed for the aspect nodes contained in the shingle (see Figure 3a). As shingles can contain overlapping set of aspects, it can represent the POIs and user preferences of overlapping aspects as well. The shingles of a user node represent the set of aspects that adhere to her preferences (the preference can be ordered based on the similarity score of a user node to the shingles). As our goal is to cluster (user, POI) tuples, we need to find the sets of user and POI nodes that share sufficiently large number of shingles. Each shingle contains the associated aspects which relates users and POIs. We can easily find the top n_u users and top n_l POIs whose similarity score is high for this shingle. The overall process can be achieved in polynomial time [6, 7] and is dependent on the number of nodes in the graph, number of shingles to use, and the size of a shingle. The normalized similarity score between a POI $_l$ and an aspect (a) from all shingles is defined as:

$$Score(l, a) = \frac{1}{|Sh|} \sum_{a \in Sh} \frac{1}{k} sim^Y(l, Sh), \quad (3)$$

where Sh is the set of ordered shingles that contain aspect a , and k is the similarity-based order of the relevant shingle. This score is interpolated to the 5-star rating scheme similar to ReEL-Core.

Finding the subsets of aspects with highest similarity score not only facilitates explanation of recommendation but also provisions clustering of users who have similar preferences on aspects (even in absence of explicit social links) and generating a group recommendation. It can also be used to generate preference wise recommendation (e.g., for the set of users $\{u_1, u_2, u_5\}$ the set of aspects {"food", "service"} might be interesting, for the set of users $\{u_1, u_2, u_3\}$ the set of aspects {"food", "price"} might be interesting, etc.). This can also facilitate the clustering of POIs that are preferred for similar aspects (e.g., the set of hotels that are popular for "Service").

4.3 Ranking Method (ReEL-Rank)

This model uses the frequency of usage of an aspect to a place. The places recommended to a user and the places' relevant aspects are used as graph nodes. The weight of a place-aspect edge indicates the overall positive opinions on the place for the aspect. A ranking function is then defined as:

$$Rank(i) = \frac{1-d}{N} + d * \sum_{(j,i) \in E} \frac{Rank(j) * W_{j,i}}{O_j}, \quad (4)$$

where $Rank(i)$ is the rank of a node i , d ($=0.85$) is the damping factor, N is number of nodes in the graph, E is set of edges in the graph, $W_{j,i}$ is weight of the edge (j, i) , and O_j is number of outgoing links from node j . The ranks are iteratively updated till the graph is converged. The highest ranking aspect node and its highest ranking neighbors give the places that are noted for this aspect. Similarly, other higher ranking aspect nodes and their neighbors are accessed to get the other place-aspect pairs. For a given aspect, the neighbor nodes are sorted based on their rank before the explanation is generated. An explanation of the following form is generated: (i) *Food*: Places ordered by rank: Place 1, Place 2, ... (ii) *Service*: Places ordered by rank: Place 4, Place 5, ..., etc. The rank of a place on an aspect is aggregated from all the users to get the star rating score.

Attributes	Yelp ¹	TripAdvisor ²	AirBnB ³
Reviews	2,225,213	246,399	570,654
Users	552,339	148,480	472,701
Places	77,079	1,850	26,734
Words	302,979,760	43,273,874	54,878,077
Sentences	18,972,604	2,167,783	284,1004
Average Sentences/review	8.53	8.79	4.98
Average Words/review	136.15	175.62	96.16
Average Reviews/user	4.03	1.66	1.20
Average Reviews/place	28.87	133.18	21.34
4, 5 stars ⁴	591,618 and 900,940	78,404 and 104,442	479,842
1, 2 stars	260,492 and 190,048	15,152 and 20,040	5,766

Table 1: Statistics of the datasets

5 EVALUATION

We defined four models: (i) **DAP** - the model that used a deep network and factorization machine for recommendations and has no provision for explanation, (ii) **ReEL-Core** - the model that used bipartite core, (iii) **ReEL-Dense** - the model that used dense subgraph extraction, and (iv) **ReEL-Rank** - the model that used a ranking approach for explanation generation. We also evaluated the Aspect extraction, Aspect categorization, Sentence-aspect classification modules in terms of accuracy.

- (1) Aspect extraction: We used the SemEval 2014 Task 4: Aspect Based Sentiment Analysis Annotation dataset as the benchmark data and were able to get an accuracy of 70.04%.
- (2) Aspect categorization: We got an accuracy of 67.12% with the SemEval 2014 Task 4: Aspect Based Sentiment Analysis Annotation dataset.
- (3) Sentence-aspect classification: We used 100, 150, and 200 epochs with 32 and 64 batches. With 200 epochs and 64 batches, we got 69.01% accuracy on Yelp dataset.

We compared the performance of our proposed models with the following models: (1) UCF [22] uses the user-based collaborative filtering technique, (2) ICF [39] uses item-based collaborative filtering, (3) PPR [20] uses personalized page ranking, (4) Guo et al. [19] uses aspect-aware POI recommendation, (5) ORec [53] uses opinion-based POI recommendation, (6) *Word-embedding approach*: In this approach, the review sentences from a user and the one for an item are mapped to a latent space using the word embedding [32]. For a user, the K -nearest neighbors in the space were considered as the top- K recommendations, (7) *Latent Dirichlet Allocation approach* [5]: In this model, we extract the topics relevant to a user and the topics relevant to places. The user-place tuples with most common topics are used for the recommendation, and (8) *DeepConn* [58]: This is the CNN-based model which uses the review embeddings but ignores the other contextual embedding and the polarity of reviews.

5.1 Dataset

We used three real-world datasets to evaluate the proposed models. Table 1 shows that in all three datasets, most of the users tend to give high (positive) ratings to the places. The top-10 terms of different aspects are illustrated in Table 3b.

Experimental settings We used a 5-fold cross validation to evaluate the models. The frequency thresholds for noun and noun

¹ https://www.yelp.com/dataset_challenge ² Wang et al. [44] ³ <http://insideairbnb.com/get-the-data.html>

⁴ explicitly missing ratings, neutral, and zero ratings are not shown

Figure 3: (a) Shingles extraction (shown without edge weights), (b) Top-terms in different categories, (c) bipartite cores

Figure 4: (a) Average performance of different models (b) Precision@N and Recall@N of different models

phrase extraction were set to 100, 250, and 500. Our experimental

analysis show better results with 100. The CNN used 128 filters, 64

batches, 200 epochs, and embedding vectors of size 384. We used an Ubuntu 14.04.5 LTS, 32 GB RAM, a Quadcore Intel(R) Core(TM) i7-3820 CPU @ 3.60 GHz machine. We used the same configuration with Tesla K20c 6 GB GPU to evaluate neural network-based models.

5.2 Experimental Results and Discussion

We used the reviews of users and places with at least five reviews. We used a 5-fold cross validation and the precision (p), recall (r), and f-score ($2 \cdot p \cdot r / (p + r)$) metrics for evaluation. We considered the top @5, @10, @15, and @20 recommended items for the evaluation. The evaluation of different models is shown in Table 4a. The Precision@N and Recall@N of different models is shown in Figure 4b.

The results show that the ICF performed least, UCF and PPR performed on par, model from Guo et al. [19] performed better than ORec [53], LDA [5], and Embedding [30] models. Among the ones without explanation, **DAP** performed best on the Yelp dataset. Though it outperformed in other two datasets as well, the difference was not significant. This implies that for larger datasets, the performance of the proposed model is outstanding. This is common with DNNs which need a reasonably large training data for better performance. The recall of DeepConn [58] was higher than that of **DAP** in the TripAdvisor dataset but its precision was lower. This might be because of the sentence-level sentiment which was exploited in **DAP** but not in DeepConn [58].

Unlike **DAP**, which provided a single list of recommendations and selected top@N POIs from the list, the **ReEL-Core** and **ReEL-Rank** produced individual lists for each aspect, and outperformed **DAP** because they categorized recommendations into different aspect categories which led to the re-ordering of the items into small recommendation lists. This re-ordering can help increase the number of true positives and decrease the false positives, as the least preferred items might move to the later part of the recommended lists and the more preferred ones move to the front part of the lists. The **ReEL-Core** outperformed **ReEL-Rank** and **ReEL-Dense**. One reason is due to the repeated bipartite core extraction by **ReEL-Core** where the nodes got re-ranked for every bipartite core but the **ReEL-Rank** only ranked all the nodes just once. After having the ordered set of places within each aspect, having an explanation of type similar to [27] (i.e. place A is better than 80% of places for "Food", etc.) can be achieved by counting the number of places behind the target place in the recommended list.

5.3 Evaluation of Explainability

For a place p , the aspect popularity of an aspect a can be defined in terms of the number of positive and negative mentions:

$$\text{AspectPopularity}(p_a) = \sum_{\text{sentence} \in \text{Review}_p} (|\text{positive}| \mid a - |\text{negative}| \mid a). \quad (5)$$

To check the presence of correct aspects in the explanation, we ordered the aspects of every place based on the aspect popularity score. We used a trigram across the extracted aspects to identify the sentiment polarity of the aspects. The relevant aspects were ordered by the aspect popularity score. So, a place can be represented by the set of aspects ordered by the popularity: $p_a = \{a_1, a_2, \dots, a_n\}$. For every explanation, we took the aspects for which a place was recommended. The aspects were ordered based on the order of cores (primary, secondary, etc.). This gave us another set of aspects for

every place. The performance of explainability was then measured in terms of Levenshtein distance between the lists. The average Levenshtein distance across all places was observed to be 20%.

5.4 Impact of Explanation- A Case Study

We analyzed the role of **ReEL-Core**, using the top-5 bipartite cores (see Table 3c) extracted for three users - "7iigQ2XM-V0ciwmCIdrIBA", "7Mg6r6g7RUwQH_Bllrd-wQ", and "9HDElil2309UajBgtYcD4w", hereafter called as u_1 and u_2 , and u_3 respectively. We can see that the ordered preferences of user u_1 are "Price", "Pet", "Service", "Food", and "Amenities". This implies the highest preference of u_1 on "Price", regardless of the order of POIs recommended.

For user u_1 , the POI "NK3S3U6TQtysH_-eqT3bBQ" was the second highly recommended place by regular recommender. With the **ReEL-Core**, it is categorized into "Others" bipartite core - *the sixth core*. If the user really cares about other cores (i.e. related to other aspect categories) then having it in sixth core is better than having it in front list. The least recommended POI "p9Bl3BxPltz2WnLxJLnBvw" by simple recommender is now categorized as the least popular item for the primary bipartite core (i.e. related to "Price"), and three other secondary cores (i.e. related to "Service", "Pet", and "Food"). Many POIs ranked in the later part of the list by the simple recommenders are found within top-20 of the different bipartite cores. Have this user used the simple recommendation, and considered only the top-20 recommendations, then these items would have been missed. A sample explanation for user u_1 is the ordered set of places taken from the ordered bipartite cores:

Recommendation: (1) Place 1, Place 2,...; **Explanation:** Popular for Price. (2) Place 3, Place 4,...; **Explanation:** Popular for Service.

Similarly, the place "v4iA8kusUrB19y2QNOiUbw" that was most recommended item for user u_2 by the simple recommender is categorized to sixth bipartite core (i.e. "Others"). The place "HxPpZSY6Q1eARuihhra6A" that did not fit in top-20 of simple recommender is found in the sixth position of first three bipartite cores. The location "mh1le9QGMrZLohAjfheJJg" which was the second least recommended by simple recommender is categorized as the second least preferred item for the first five bipartite core (i.e. "Service", "Price", "Pet", "Food", and "Amenities"). A similar analysis observed for 500 other users is skipped due to space constraint.

6 CONCLUSION AND FUTURE WORK

We formulated user-aspect bipartite relation as a bipartite graph and exploited bipartite-core, shingles, and ranking-based techniques to predict the ordered aspect preferences of users for explainable recommendation. The proposed models supplemented with explanations outperformed the ones without explanation, and gained significant improvement (e.g., 5.8% to 29.5% from DeepConn [58], and 11.1% to 27.4% from Guo et al. [19]) on F-score over relevant studies. In future, we would like to exploit different aspect extraction techniques, cluster the users based on their preference order on aspect categories, and generate group recommendations.

ACKNOWLEDGEMENT

This research is partially supported by US Army Research Lab under the grant number W911NF-12-R-0012.

REFERENCES

- [1] Ramesh Baral and Tao Li. 2016. MAPS: A multi aspect personalized poi recommender system. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 281–284.
- [2] Ramesh Baral and Tao Li. 2017. Exploiting the roles of aspects in personalized POI recommender systems. *Data Mining and Knowledge Discovery* (2017), 1–24.
- [3] Ramesh Baral, Dingding Wang, Tao Li, and Shu-Ching Chen. 2016. Geotecs: exploiting geographical, temporal, categorical and social aspects for personalized poi recommendation. In *Information Reuse and Integration (IRI), 2016 IEEE 17th International Conference on*. IEEE, 94–101.
- [4] Mustafa Bilgic and Raymond J Mooney. 2005. Explaining recommendations: Satisfaction vs. promotion. In *Beyond Personalization Workshop, IUI*, Vol. 5. 153.
- [5] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.
- [6] Andrei Z Broder, Moses Charikar, Alan M Frieze, and Michael Mitzenmacher. 2000. Min-wise independent permutations. *J. Comput. System Sci.* 60, 3 (2000), 630–659.
- [7] Andrei Z Broder, Steven C Glassman, Mark S Manasse, and Geoffrey Zweig. 1997. Syntactic clustering of the web. *Computer Networks and ISDN Systems* 29, 8-13 (1997), 1157–1166.
- [8] Annalina Caputo, Pierpaolo Basile, Marco de Gemmis, Pasquale Lops, Giovanni Semeraro, and Gaetano Rossiello. 2017. SABRE: A Sentiment Aspect-Based Retrieval Engine. In *Information Filtering and Retrieval*. Springer, 63–78.
- [9] Guanliang Chen and Li Chen. 2015. Augmenting service recommender systems by incorporating contextual opinions from user reviews. *User Modeling and User-Adapted Interaction* 25, 3 (2015), 295–329.
- [10] Li Chen and Feng Wang. 2017. Explaining recommendations based on feature sentiments in product reviews. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces*. ACM, 17–28.
- [11] Xu Chen, Zheng Qin, Yongfeng Zhang, and Tao Xu. 2016. Learning to rank features for recommendation over multiple categories. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 305–314.
- [12] Jiajun Cheng, Shenglin Zhao, Jiani Zhang, Irwin King, Xin Zhang, and Hui Wang. 2017. Aspect-level Sentiment Classification with HEAT (HiErarchical ATtention) Network. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, 97–106.
- [13] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research* 12, Aug (2011), 2493–2537.
- [14] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 191–198.
- [15] Ruihai Dong and Barry Smyth. 2017. User-based Opinion-based Recommendation. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. AAAI Press, 4821–4825.
- [16] Christiane Fellbaum. 1998. *WordNet*. Wiley Online Library.
- [17] Fatih Gedikli, Mouzhi Ge, and Dietmar Jannach. 2011. Understanding recommendations by reading the clouds. In *International Conference on Electronic Commerce and Web Technologies*. Springer, 196–208.
- [18] Fatih Gedikli, Dietmar Jannach, and Mouzhi Ge. 2014. How should I explain? A comparison of different explanation types for recommender systems. *International Journal of Human-Computer Studies* 72, 4 (2014), 367–382.
- [19] Qing Guo, Zhu Sun, Jie Zhang, Qi Chen, and Yin-Leng Theng. 2017. Aspect-aware Point-of-Interest Recommendation with Geo-Social Influence. In *Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization*. ACM, 17–22.
- [20] Taher H Haveliwala. 2002. Topic-sensitive pagerank. In *Proceedings of the 11th international conference on World Wide Web*. ACM, 517–526.
- [21] Xiangnan He, Tao Chen, Min-Yen Kan, and Xiao Chen. 2015. Trirank: Review-aware explainable recommendation by modeling aspects. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM, 1661–1670.
- [22] Jonathan L Herlocker, Joseph A Konstan, Al Borchers, and John Riedl. 1999. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 230–237.
- [23] Jonathan L Herlocker, Joseph A Konstan, and John Riedl. 2000. Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*. ACM, 241–250.
- [24] Yohan Jo and Alice H Oh. 2011. Aspect and sentiment unification model for online review analysis. In *Proceedings of the fourth ACM international conference on Web search and data mining*. ACM, 815–824.
- [25] Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* (2014).
- [26] Jon M Kleinberg. 1999. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)* 46, 5 (1999), 604–632.
- [27] Aonghus Lawlor, Khalil Muhammad, Rachael Rafter, and Barry Smyth. 2015. Opinionated explanations for recommendation systems. In *Research and Development in Intelligent Systems XXXII*. Springer, 331–344.
- [28] Shoushan Li, Rongyang Wang, and Guodong Zhou. 2012. Opinion target extraction using a shallow semantic parsing framework. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*. AAAI Press, 1671–1677.
- [29] Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *ACL (System Demonstrations)*. 55–60.
- [30] Jarana Manotumruksa, Craig Macdonald, and Iadh Ounis. 2016. Modelling user preferences using word embeddings for context-aware venue recommendation. *arXiv preprint arXiv:1606.07828* (2016).
- [31] Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*. ACM, 165–172.
- [32] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.
- [33] Samaneh Moghaddam and Martin Ester. 2010. Opinion digger: an unsupervised opinion miner from unstructured product reviews. In *Proceedings of the 19th ACM international conference on Information and knowledge management*. ACM, 1825–1828.
- [34] Samaneh Moghaddam and Martin Ester. 2011. ILDA: interdependent LDA model for learning latent aspects and their ratings from online product reviews. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. ACM, 665–674.
- [35] Khalil Ibrahim Muhammad, Aonghus Lawlor, and Barry Smyth. 2016. A live-user study of opinionated explanations for recommender systems. In *Proceedings of the 21st International Conference on Intelligent User Interfaces*. ACM, 256–260.
- [36] Cataldo Musto, Fedelucio Narducci, Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. 2016. Explod: A framework for explaining recommendations based on the linked open data cloud. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 151–154.
- [37] Steffen Rendle. 2012. Factorization machines with libfm. *ACM Transactions on Intelligent Systems and Technology (TIST)* 3, 3 (2012), 57.
- [38] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*. AUAI Press, 452–461.
- [39] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*. ACM, 285–295.
- [40] Panagiotis Symeonidis, Alexandros Nanopoulos, and Yannis Manolopoulos. 2008. Providing justifications in recommender systems. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* 38, 6 (2008), 1262–1272.
- [41] Nava Tintarev and Judith Masthoff. 2012. Evaluating the effectiveness of explanations for recommender systems. *User Modeling and User-Adapted Interaction* 22, 4 (2012), 399–439.
- [42] Ivan Titov and Ryan T McDonald. 2008. A Joint Model of Text and Aspect Ratings for Sentiment Summarization. In *ACL*, Vol. 8. Citeseer, 308–316.
- [43] Jesse Vig, Shilad Sen, and John Riedl. 2009. Tagsplanations: explaining recommendations using tags. In *Proceedings of the 14th international conference on Intelligent user interfaces*. ACM, 47–56.
- [44] Hongning Wang, Yue Lu, and ChengXiang Zhai. 2011. Latent aspect rating analysis without aspect keyword supervision. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 618–626.
- [45] Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. 2017. Coupled Multi-Layer Attentions for Co-Extraction of Aspect and Opinion Terms. In *AAAI*. 3316–3322.
- [46] Xiangyu Wang, Yi-Liang Zhao, Liqiang Nie, Yue Gao, Weizhi Nie, Zheng-Jun Zha, and Tat-Seng Chua. 2015. Semantic-based location recommendation with multimodal venue semantics. *IEEE Transactions on Multimedia* 17, 3 (2015), 409–419.
- [47] Yequan Wang, Minlie Huang, Li Zhao, and others. 2016. Attention-based lstm for aspect-level sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 606–615.
- [48] Bin Xia, Tao Li, Qianmu Li, and Hong Zhang. 2018. Noise-tolerance matrix completion for location recommendation. *Data Mining and Knowledge Discovery* 32, 1 (2018), 1–24.
- [49] Bin Xia, Zhen Ni, Tao Li, Qianmu Li, and Qifeng Zhou. 2017. Vrer: context-based venue recommendation using embedded space ranking SVM in location-based social network. *Expert Systems with Applications* 83 (2017), 18–29.
- [50] Dingqi Yang, Daqing Zhang, Zhiyong Yu, and Zhu Wang. 2013. A sentiment-enhanced personalized location recommendation system. In *Proceedings of the 24th ACM Conference on Hypertext and Social Media*. ACM, 119–128.
- [51] Quan Yuan, Gao Cong, Zongyang Ma, Aixin Sun, and Nadia Magnenat Thalmann. 2013. Time-aware point-of-interest recommendation. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information*

- retrieval*. ACM, 363–372.
- [52] Zhongwu Zhai, Bing Liu, Hua Xu, and Peifa Jia. 2011. Clustering product features for opinion mining. In *Proceedings of the fourth ACM international conference on Web search and data mining*. ACM, 347–354.
 - [53] Jia-Dong Zhang, Chi-Yin Chow, and Yu Zheng. 2015. ORec: An opinion-based point-of-interest recommendation framework. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM, 1641–1650.
 - [54] Lei Zhang and Bing Liu. 2014. Aspect and entity extraction for opinion mining. In *Data mining and knowledge discovery for big data*. Springer, 1–40.
 - [55] Wei Zhang, Quan Yuan, Jiawei Han, and Jianyong Wang. 2016. Collaborative Multi-Level Embedding Learning from Reviews for Rating Prediction.. In *IJCAL* 2986–2992.
 - [56] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. 2014. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. ACM, 83–92.
 - [57] Kaiqi Zhao, Gao Cong, Quan Yuan, and Kenny Q Zhu. 2015. SAR: A sentiment-aspect-region model for user preference analysis in geo-tagged reviews. In *Data Engineering (ICDE), 2015 IEEE 31st International Conference on*. IEEE, 675–686.
 - [58] Lei Zheng, Vahid Noroozi, and Philip S Yu. 2017. Joint Deep Modeling of Users and Items Using Reviews for Recommendation. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM, 425–434.