Hotel2vec: Learning Attribute-Aware Hotel Embeddings with Self-Supervision

Abstract

We propose a neural network architecture for learning vector representations of hotels. Unlike previous works, which typically only use user click information for learning item embeddings, we propose a framework that combines several sources of data, including user clicks, hotel attributes (e.g., property type, star rating, average user rating), amenity information (e.g., the hotel has free Wi-Fi or free breakfast), and geographic information. During model training, a joint embedding is learned from all of the above information. We show that including structured attributes about hotels enables us to make better predictions in a downstream task than when we rely exclusively on click data. We train our embedding model on more than 40 million user click sessions from a leading online travel platform and learn embeddings for more than one million hotels. Our final learned embeddings integrate distinct sub-embeddings for user clicks, hotel attributes, and geographic information, providing an interpretable representation that can be used flexibly depending on the application. We show empirically that our model generates high-quality representations that boost the performance of a hotel recommendation system in addition to other applications. An important advantage of the proposed neural model is that it addresses the cold-start problem for hotels with insufficient historical click information by incorporating additional hotel attributes which are available for all hotels.

Introduction

Learning semantic representations (embeddings) of different entities, such as textual, commercial, and physical, has been a recent and active area of research. Such representations can facilitate applications that rely on a notion of similarity, for example recommendation systems and ranking algorithms in e-commerce.

In natural language processing, word2vec BIBREF0 learns vector representations of words from large quantities of text, where each word is mapped to a d-dimensional vector such that semantically similar words have geometrically closer vectors. This is achieved by predicting either the context words appearing in a window around a given target word (skip-gram model), or the target word given the context (CBOW model). The main assumption is that words appearing frequently in similar contexts share statistical properties (the distributional hypothesis). Crucially, word2vec models, like many other word embedding models, preserve sequential information encoded in text so as to leverage word co-occurrence statistics. The skip-gram model has been adapted to other domains in order to learn dense representations of items other than words. For example, product embeddings in e-commerce BIBREF1 or vacation rental embeddings in the hospitality domain BIBREF2 can be learned by treating purchase histories or user click sequences as sentences and applying a word2vec approach.

Most of the prior work on item embedding exploit the co-occurrence of items in a sequence as the main signal for learning the representation. One disadvantage of this approach is that it fails to incorporate rich structured information associated with the embedded items. For example, in the travel domain, where we seek to embed hotels and other travel-related entities, it could be helpful to encode explicit information such as user ratings, star ratings, hotel amenities, and location in addition to implicit information encoded in the click-stream.

In this work, we propose an algorithm for learning hotel embeddings that combines sequential user click information in a word2vec approach with additional structured information about hotels. We propose a neural architecture that adopts and extends the skip-gram model to accommodate arbitrary relevant information of embedded items, including but not limited to geographic information, ratings, and item attributes. In experimental results, we show that enhancing the neural network to jointly encode click and supplemental structured information outperforms a skip-gram model that encodes the click information alone. The proposed architecture also naturally handles the cold-start problem for hotels with little or no historical clicks. Specifically, we can infer an embedding for these properties by leveraging their supplemental structured metadata.

Compared to previous work on item embeddings, the novel contributions of this paper are as follows:

We propose a novel framework for fusing multiple sources of information about an item (such as user click sequences and item-specific information) to learn item embeddings via self-supervised learning.

We generate an interpretable embedding which can be decomposed into sub-embeddings for clicks, location, ratings, and attributes, and employed either as separate component embeddings or a single, unified embedding.

It is also dynamic, meaning it is easy to reflect future changes in attributes such as star-rating or addition of amenities in the embedding vectors without retraining.

We address the cold-start problem by including hotel metadata which are independent of user click-stream interactions and available for all hotels. This helps us to better impute embeddings for sparse items/hotels.

We show significant gains over previous work based on click-embedding in several experimental studies.

The structure of the remainder of this paper is as follows. Section 2 gives an overview of some of the recent works on neural embedding. Section 3 provides details of the proposed framework, including the neural network architecture, training methodology, and how the cold-start problem is addressed. In Section 4, we present experimental results on several different tasks and a comparison with previous state-of-the-art work. Section 5 concludes the paper.

Related Work

Recommendation is an inherently challenging task that requires learning user interests and behaviour. There has been a significant body of research on advancing it using various frameworks BIBREF3, BIBREF4, BIBREF5, BIBREF6, BIBREF7. Learning a semantic representation/embedding of the items being recommended is a critical piece of most of these frameworks.

Deep learning models have been widely used for learning embeddings BIBREF8, BIBREF9, BIBREF10. One prominent use case is learning product embeddings for e-commerce. In BIBREF1, BIBREF11, the authors develop an approach based on the skip-gram model BIBREF0, frequently used in natural language processing. They leverage users' purchase histories obtained from their e-mail receipts to learn a dense representation of products. Each user's complete purchase history is represented as a sequence, which is treated as a sentence in which the items are considered words.

In more recent work BIBREF2, the authors use the skip-gram framework to learn embeddings for vacation rental properties. They extend the ideas in BIBREF1 to take into account a user's click stream data during a session. A key contribution of their method is the modification of the skip-gram model to always include the booked hotels in the context of each target token, so that special attention is paid to bookings. They also improve negative sampling by sampling from the same market, which leads to better within-market listing similarities. Nevertheless, their model relies exclusively on large amounts of historical user engagement data, which is a major drawback when such data are sparse.

In another relevant work, BIBREF12, authors propose a framework for YouTube video recommendation which fuses multiple features (e.g., video watches, search tokens, geo embeddings) into a unified representation via a neural architecture. They then use these embeddings for candidate generation and ranking. The main limitation of this work is that the individual embeddings are learned separately, and then combined via a neural network to perform classification.

Similar to our work on hotel2vec, there are also some works which attempt to include explicit item attributes (e.g., size, artist, model, color) within the sequence prediction framework using various strategies. In BIBREF13, the item metadata is injected into the model as side information to regularize the item embeddings. In their approach, they only use one feature (singer ID) in the experiments. In addition, their approach does not accommodate learning independent embedding vectors for each attribute group. Most recently, BIBREF14 propose a method where they train separate encoders for text data, click-stream session data, and product image data, and then use a simple weighted average to unify these embeddings. The weights are learned using grid search on the downstream task. While their approach allows for exploring independent embedding vectors, the sub-embeddings of different attribute groups are learned independently rather than jointly. In addition to efforts extending the skip-gram framework, emerging research attempts to extend GloVe BIBREF15 by incorporating various attributes. BIBREF16 incorporate attribute information into GloVe by modifying the loss function such that the representation of a location can be learned by combining both text and structural data.

The Proposed Framework

Similar to BIBREF0, by treating the clicks made by users within an interactive web session as words, and sequences of clicks as sentences, we seek to predict the context hotels (words), given a target hotel (word) in the session (sentence). On a high level, this is the approach proposed in BIBREF1, BIBREF2. We refer to this approach as a session-only model.

As mentioned earlier, one drawback of this approach is that it does not use any information apart from the click data, making it very challenging to make predictions for unseen hotels or hotels with sparse click data. In addition, the model may be forced to learn certain semantic features which capture aspects of user interest, hotel geographic information, hotel attributes, and so on, as latent variables as opposed to leveraging them as explicitly-provided input features. To address these shortcomings, we propose adding more explicit information about the hotel as model input. Intuitively, this should make the model more efficient during training as well as provide information that it can use when making predictions on unseen or sparse hotels.

Another major advantage of our model is its use of different projection layers for various hotel/item attributes. This enables us to learn independent embedding vectors representing different facets of the property, in addition to an enriched, unified embedding for each hotel. This model also provides a dynamic framework for updating the embedding of a hotel, once its user-rating or other attribute information changes over time. This is not trivial in session-only models, unless we re-train a new model based on recent click data post attribute changes. In the remainder of the paper, we refer to our proposed model as an enriched model, in contrast to the session-only model introduced above.

The Proposed Framework ::: Neural Network Architecture

Figure FIGREF7 illustrates the proposed architecture for an enriched, hotel2vec model. As we can see, each aspect of the hotel is embedded separately, and these representations are later concatenated and further compressed before being used for context prediction.

Formally, a click session is defined as a sequence of hotels (items) $\lbrace h_1, h_2, \cdots , h_n\rbrace $ clicked on by a user during a defined window of time or visit. We denote the click, amenity, geographic, and enriched embedding vectors with $\mathbf {V}_c$, $\mathbf {V}_a$, $\mathbf {V}_g$, and $\mathbf {V}_e$ respectively. These are defined as follows:

where I_c is the one-hot encoding of hotels in the click session, and I_g is a continuous vector with geographical coordinates of the hotel. Amenity features can be categorical or numerical with possible missing values. Thus, I_a is partitioned per feature, where for numerical features we simply use an element of I_a assigned with the value of that feature, and for categorical features with m categories, we assign m elements of I_a and set the corresponding category to 1 and the others to 0. If the feature is missing, we set everything to 0. $\operatornamewithlimits{ReLU}$ is the rectified linear unit activation function BIBREF17 and $f(x; \mathbf {W})$ is a normalized projection layer parameterized with trainable weights $\mathbf {W}$, i.e., $f(x; \mathbf {W}) = \operatornamewithlimits{ReLU}(\frac{x \mathbf {W}}{\hphantom{{}_2}\Vert x \mathbf {W} \Vert _{\scriptstyle {2}}})$.

We train our model using negative sampling based on optimizing the noise contrastive estimation (NCE) loss BIBREF18. More formally, given h_t as the target, we estimate the probability of h_c being a context hotel to be

where $\mathbf {W}_{c,:}$ is the c^{th} row of $W_{\scriptstyle _{NCE}}$. We find parameters of the model by maximizing the probabilities of correct predictions. We train the model using backpropagation and minimizing the following loss function:

where $\mathbf {V}_{e_{\scriptstyle {t}}}$ is the enriched embedding of h_t, $\mathbf {W}_{i,:}$ is i^{th} row of $W_{\scriptstyle _{NCE}}$ matrix, $\mathcal {N}_c = \lbrace h_i| 1 \le i \le N, h_i \sim P_n(h_c)\rbrace $ is the set of negative examples, and $P_n(h_c)$ is the distribution which we use to pick the negative samples. We train our model by maximizing equation DISPLAY_FORM10 using batch stochastic gradient descent.

The Proposed Framework ::: Negative Sampling

It is well known BIBREF18, BIBREF0, BIBREF19 that using negative sampling, a version of noise contrastive estimation, significantly decreases the amount of time required to train a classifier with a large number of possible classes. In the case of recommendation, there is typically a large inventory of items available to recommend to the user, and thus we train our skip-gram model using negative sampling. However, it is not uncommon that users frequently search exclusively within a particular subdomain. For example, in hotel search, a customer looking to stay in Miami will focus on that market and rarely across different markets. This motivates a more targeted strategy when selecting negative samples: we select half of our negative samples following the schema in BIBREF20, i.e., from the complete set of all hotels, and the other half uniformly at random from the same market as the clicked hotel. Throughout this paper, a market is defined as a set of similar hotels in the same geographic region. It's worth noting that there may be multiple markets in the same city or other geo region. In the experimental section, we show that this improves the model's within-market similarities and its predictions.

The Proposed Framework ::: Cold Start Problem

In practice, many hotels/items appear infrequently or never in historical data. Recommender systems typically have difficulty handling these items effectively due to the lack of relevant training data. Apart from the obvious negative impacts on searchability and sales, neglecting these items can introduce a feedback loop. That is, the less these items are recommended, or the more they are recommended in inappropriate circumstances, the more the data reinforces their apparent lack of popularity.

Dealing with such hotels/items and choosing appropriate weights for them is referred to as the "cold start problem." One of the main advantages of the enriched hotel2vec model over session-only approaches is its ability to better handle cold start cases. Although an item might lack sufficient prior user engagement, there are often other attributes available. For example, in our use case, thousands of new properties are added to the lodging platform's inventory each quarter. While we don't have prior user engagement data from which to learn a click embedding, we do have other attributes such as geographical location, star rating, amenities, etc. Hotel2vec can take advantage of this supplemental information to provide a better cold-start embedding.

Experimental Results

In this section, we present several experiments to evaluate the performance of the trained hotel2vec embeddings. Before diving into the details of the experiments, we first describe the dataset and model parameters.

Experimental Results ::: Experimental Framework

Our dataset contains more than 40M user click sessions, which includes more than 1.1 million unique hotels. A click session is defined as a span of clicks performed by a user with no gap of more than 7 days. We randomly split the sessions into training, validation, and test with a ratio of 8:1:1.

We use a system with 64GB RAM, 8 CPU cores, and a Tesla V100 GPU. We use Python 3 as the programming language and the Tensorflow BIBREF21 library for the neural network architecture and gradient calculations. is sufficient for prevention of overfitting.

We tune the hyperparameters for all models, including the baseline session-only model, on the validation set. We search for a learning rate from $\lbrace 0.01, 0.1, 0.5, 1.0, 2.5\rbrace $ and embedding dimensions from $\lbrace 32, 128\rbrace $. To train the model weights, we use stochastic gradient descent (SGD) with exponential decay since it performs better than other optimizers in our case, and a batch size of 4096.

For our implementation of the session-only model, a learning rate of 0.5 and embedding dimension of 32 worked best. Throughout the remainder of the paper, we refer to this model as the session-32 model. For our enriched model (hotel2vec), a learning rate of 0.05 worked best; for the dimensionality of the embedding vectors, we found that letting $V_c, V_e \in {R}^{32}$, $V_a \in {R}^{15}$ and $V_g \in {R}^{5}$ worked best. We refer to this model as the enriched-32 model.

Experimental Results ::: Quantitative Analysis

Experimental Results ::: Quantitative Analysis ::: Hits@k for hotel context prediction

A robust metric for evaluating a set of hotel embeddings (or, more generally, any set of items displayed to a user in response to an information need) is its ability to predict a user's next click/selection. In this section, we compare our model based on the hits@k metric in various scenarios. Hits@k measures the average number of times the correct selection appears in the top k predictions.

We consider two main scenarios: in the first, we are given the current hotel clicked by the user, and we try to predict the next clicked hotel among all approximately 1.1M hotels (raw evaluation). The second scenario is identical except we limit the candidates to hotels within the same market (filtered evaluation).

Table TABREF19 shows hits@k for $k \in \lbrace 10, 100, 1000\rbrace $ for both the Session-32 and Enriched-32 models. The enriched model outperforms the session-only model by a huge margin, demonstrating the utility of including item attributes when learning embeddings. We also compare both models in the filtered scenario. This is a more realistic case because limiting hotels to the same market reduces the effect of other information the recommender system can use to provide more relevant suggestions to the user. Table TABREF19 shows predictions results in the filtered scenario.

As demonstrated by Table TABREF19, the enriched model outperforms the baseline session model significantly in both scenarios. This shows the effectiveness of hotel2vec in incorporating both click sessions and item/hotel attributes for better recommendations.

Experimental Results ::: Quantitative Analysis ::: Comparison using cosine similarity

In this section, rather than using the model's output probabilities to induce a ranking over hotels, we measure hits@k over the ranking induced using cosine similarity of the embedding vectors. This is useful in scenarios where it isn't feasible to directly use the model's probabilities. Table TABREF21 shows the results for various embeddings. We show that using the enriched vectors one achieves the highest performance.

We also see from Table TABREF21 that using cosine similarity instead of the whole network does not result in a huge decrease in performance. Finally, Table TABREF21 also shows that even the standalone click vectors obtained from the enriched model outperform the embeddings obtained from the session-only model.

Experimental Results ::: Quantitative Analysis ::: Average intra/inter market embedding similarities

We expect hotels in the same market to be more similar to each other than to hotels in other markets. To evaluate how well this market-level information is encoded by the learned embeddings, we calculate the average similarity between pairs of markets, with the expectation that we should see a strong diagonal component in the similarity matrix. We note that our model is not explicitly trained to learn this kind of market information. However, it is able to learn this by combining the click sessions and hotel attribute information. Figure FIGREF13 shows the average similarity scores between hotels in multiple famous cities using two of the embedding vectors. As Figure FIGREF13 clearly depicts, there is a strong similarity between hotels of the same city. Also, markets that are closer to each other (all US cities vs European vs Asian), or for reasons other than geographic proximity are expected to be more similar (e.g., Las Vegas and Macao, or Tokyo and Paris) do indeed have a higher similarity. For comparison, Figure FIGREF13 shows the average cosine similarity between and within markets for the session-only model embeddings. This model captures within-market similarity well but is not as effective as the enriched model for capturing cross-market similarity. For instance, the session-only model fails to recover the similarity between Las Vegas and Macao.

Experimental Results ::: Qualitative Analysis

The learned hotel embeddings can be used for recommending similar hotels in various situations. In this section, we show examples of how these embeddings are helpful with real examples of hotels from our dataset.

Experimental Results ::: Qualitative Analysis ::: Visualization of embeddings

To further illuminate the nature of the embeddings learned by the hotel2vec model, we examine a low-dimensional projection of hotel embeddings in the Miami market (Figures FIGREF25 and FIGREF25). The colors signify the grouping of hotels into various competing subcategories (i.e., similar hotels), manually annotated by a human domain expert. The enriched model is significantly better at clustering similar hotels than the session-only model.

Experimental Results ::: Qualitative Analysis ::: Most similar hotels

A common scenario is finding similar hotels to a target hotel in other destinations. For example, when the user searches for a specific hotel name (e.g., Hotel Beacon, NY) we would like to be able to recommend a few similar hotels. The learned embeddings can be used to find top-k most similar hotels to a given one. Given a target hotel h, we compute the cosine similarity of every other hotel with h and pick the most similar hotels. Rigid evaluation of this system requires A/B testing; here we show a few examples comparing our enriched embeddings and the session-only embeddings in Figure FIGREF29 to provide some intuition for the behavior of the two models.

Experimental Results ::: Qualitative Analysis ::: Algebraic operations on hotel embeddings

We also investigate whether we can perform meaningful algebraic operations on trained hotel embeddings (similar to the semantic analogy task in BIBREF0). We pose the question "h_1 is to h_2 as h_3 is to h_x" and find h_x as the hotel with the closest vector to $\mathbf {V_{e_1}}-\mathbf {V_{e_2}}+\mathbf {V_{e_3}}$. Figure FIGREF31 shows an example of such analogy. h_1 is a Marriott hotel in NY, h_2 is a Hilton in NY, and h_3 is a Marriott in LA (near airport). The obtained h_x, is a Hilton hotel in LA near the airport, showing the amount of information captured by the enriched embeddings.

Experimental Results ::: Addressing the Cold Start Problem

Here we analyze how well the model learns embeddings for hotels with little to no presence in the training data. To demonstrate the effectiveness of our model, we compare the enriched model's hits@k with the session-only model's hits@k, for 14K target hotels that were absent during training. Table TABREF33 shows results in the filtered scenario. As we can see, the proposed enriched embedding significantly outperforms the session based embeddings for cold-start hotels.

In addition, we use a simple heuristic for cold-start imputation and compare the results with the enriched model for cold-start hotels. To impute vectors for cold-start hotels, we borrow the idea in BIBREF2 and use price, star rating, geodesic distance, type of the property (e.g., hotel, vacation rental, etc.) size in terms of number of rooms, and the geographic market information. For each imputed property, we collect the most similar properties in the same market based on the above features, considering only those properties that fall within a radius of 5km of the target hotel. Results are in Table TABREF33. The heuristic imputation technique improves the Session-32 model's performance on cold-start hotels, but it remains well below that of the enriched model.

Experimental Results ::: Training Convergence Analysis

In this section, we first look at the learning curves for both the session-32 and enriched-32 models. Then, we analyse the effect of N (number of negative samples), lr (learning rate), and the optimization algorithm on the performance of our model.

Figure FIGREF35 shows the overall training progress of both the session-32 and enriched-32 models with their respective best hyperparameters. As shown in Figure FIGREF35, our model achieves similar performance with fewer data.

An interesting phenomenon is the effect of increasing the number of negative samples on training time and accuracy. Although it takes more time to create a large number of negative samples, as Figure FIGREF36 shows, using more negative samples results in faster training times.

We show empirical experiments with various optimization algorithms and learning rates, summarized in

Figure FIGREF37. Surprisingly, we see that SGD with exponential learning rate decay outperforms most optimizers with sophisticated learning rate adaptations. We believe this is due to large variance and overfitting in the early stages of training. These issues have been observed in other tasks such as BIBREF22, BIBREF23, BIBREF24, suggesting the need to use tricks such as warm-up heuristics when using momentum-based optimization algorithms to learn embeddings on large, diverse datasets such as ours.

Conclusion

In this work, we propose a framework to learn a semantic representation of hotels by jointly embedding hotel click data, geographic information, user rating, and attributes (such as stars, whether it has free breakfast, whether pets are allowed, etc.). Our neural network architecture extends the skip-gram model to accommodate multiple features and encode each one separately. We then fuse the sub-embeddings to predict hotels in the same session. Through experimental results, we show that enriching the neural network with supplemental, structured hotel information results in superior embeddings when compared to a model that relies solely on click information. Our final embedding can be decomposed into multiple sub-embeddings, each encoding the representation for a different hotel aspect, resulting in an interpretable representation. It is also dynamic, in a sense that if one of the attributes or user ratings changes for a hotel, we can feed the updated data to the model and easily obtain a new embedding. Although we mainly focus on learning embeddings for hotels, the same framework can be applied to general item embedding, such as product embedding on Amazon, Ebay, or Spotify.

Acknowledgments

The authors would like to thank Ion Lesan, Peter Barszczewski, Daniele Donghi, Ankur Aggrawal for helping us collecting hotel's attribute, click and geographical data. We would also like to thank Dan

Friedman and Thomas Mulc for providing useful comments and feedback.