Where Should I Go? A Deep Learning Approach To Personalize Type-based Facet Ranking for POI Suggestion

Esraa Ali[1] ✉ [0000−0003−1600−3161], Annalina Caputo[2] ✉ [0000−0002−7144−8545], Séamus Lawless[1] [0000−0001−6302−258X], and Owen Conlan[1] [0000−0002−9054−9747]

[1] ADAPT Centre, School of Computer Science and Statistics, Trinity College Dublin
esraa.ali,seamus.lawless,owen.conlan@adaptcentre.ie
[2] ADAPT Centre, School of Computing, Dublin City University
annalina.caputo@adaptcentre.ie

Abstract. In a faceted search system, type-based facets (t-facets) represent the categories of the resources being searched. Ranking algorithms are needed to select and promote the most relevant t-facets. However, as these are extracted from large multi-level taxonomies, they are impossible to show entirely to the user. Facet ranking is usually employed to filter out irrelevant facets for the users. Existing facet ranking methods neglect both the hierarchical structure of t-facets and the user historical preferences. This research introduces a personalized t-facet ranking that addresses both issues. During a first step, a Deep Neural Network (DNN) model is trained to assign a relevance score to each t-facet based on three groups of relevance features. The score reflects the t-facet relevance to the user, the input query, and its general importance in the dataset. Subsequently, these scores are aggregated and the t-facets are re-organised into a smaller sub-tree to be presented to the user. Our approach aims at minimizing the effort required by the user to reach their intended search target. This is measured in terms of number of clicks the user has to perform on the t-facet tree to reach a relevant resource. The approach is applied to a Point-Of-Interest suggestion task. We solve the problem by ranking the categories of the venues as t-facets. The evaluation compares our DNN-based approach with other existing baselines and investigates the individual contribution of each group of features. Our experiment has demonstrated that the proposed personalized deep learning model leads to better t-facet rankings and minimized user effort.

Keywords: Facet Ranking · Deep Neural Networks · Personalization.

1 Introduction

The problem of Point-of-Interest (POI) suggestion has gained lot of research attention recently due to the spread of Location Based Social Networks (LBSN). The problem is concerned with recommending interesting places for the users to visit. POI suggestion algorithms personalize the recommendations according to the current user's context as well as the recorded user's history.

In this research, we solve the POI suggestion problem in the context of Faceted Search Systems (FSS), where the categories of the POIs are used as type-based facets (t-facets) that help the user to navigate the information space. Literature showed that the categories of POIs play a key role in solving this problem [2, 3]. FSS provide users with a set of t-facets to help them in filtering and narrowing down search results and locating the intended POI quickly. When POIs' categories are derived from large, multi-level taxonomies, the FSS need to implement methods to identify and prioritise the most relevant t-facet to show to the users to avoid overwhelming them.

In this work, we focus on analysing the role of personalization in t-facet ranking in isolation from other FSS aspects. We aim at answering the following research question: To what extent a Deep Neural Network (DNN) model can learn to rank t-facets in order to minimize user effort to reach the search target?

This study contributes to the research in this area by introducing a novel ranking algorithm for type-based facets relying on a deep neural network. It uses a combination of collaborative filtering (CF), query relevance, and personalization features to rank the facets. The CF features reflect the general importance of the t-facet among users. The query relevance derives the relevance of t-facets to the query from the relevance of the resources to which they belong. Finally, the personalization features exploit the user's past preferences to build a vector which represents the user interests. The extracted features are fed into a DNN model. This is trained to combine these features into a t-facet ranking score. In a following step, the approach provides a t-facet construction strategy to decide the final tree to be portrayed to the searcher.

2 Facet Ranking Related Research

Several approaches have been proposed in literature to solve the problem of personalized facet ranking, which make use of individual user models, collaborative filtering, or a mixture between the two. Factic [11] is a FSS that personalizes by building user models from semantic usage logs. Several layers of user adaptation are implemented and integrated with different weights to enhance the facet relevance model. Koren et al. [9] suggested a CF approach by leveraging explicit user feedback about the facets, which is used to build a facet relevance model for individuals. They also use the aggregated ratings to build a collaborative model for the new users in order to provide initial good facets in absence of a user profile. A personalized ranking based on CF methods was suggested by Chantamunee et. al [5, 4]. They used user ratings and Matrix factorization with SVM and Autoencoders to predict facet ranks. The Adaptive Twitter search system generates user models from Twitter to personalize facet-values ordering [1]. The user model contains entities extracted from the user's tweets. The facet-values are weighted higher if they exist in the user profile. Le et al. [10] also collect user profiles from social networks. The profile is learned from user activities and preferences using a TF-IDF feature vector model. Important facets are highlighted through matching with the vector model. All the approaches discussed so far use the same strategy to rank all types of facets. We believe it is important to distinguish between the types of facets during the ranking process as they support the user in different ways in finding their intended target. Ali et al. [8] proposed a probabilistic model to personalize t-facet ranking. Topic-based user profiles are collected from users' historical interactions with the system. A recent approach utilized Rocchio formula to build a vector representing the user interests for t-facet ranking [7]. In this model, the user's profile is expressed in a category space through vectors that capture the users' past preferences. The BERT embeddings are used as t-facet representation in vector space. The t-facet score is the cosine similarity between its BERT vector and the user profile vector.

3 Proposed Approach

Our approach to t-facet ranking is based on the intuition there are multiple 'signals' surrounding the user's interaction with the system that capture the relevance of a given t-facet. We define then the problem in terms of learning to rank t-facets based on a number of features, each capturing a different relevance aspect. A Deep Neural Network model is trained over these features, to capture the intricacy of the relationship between these features and user's profile.

When the user inputs a search query, the underlying search engine retrieves a set of relevant places.We assume that the set of retrieved results are relevant, and accordingly also the set of t-facets linked to them. This set of t-facets is the input to our ranking approach. Considering the t-facet hierarchy, the proposed approach starts by generating features for each leaf t-facet node. Then, a trained DNN model predicts a score which reflects the relevance of t-facets from user, query and collaborative perspective. Using the generated t-facet score, the second and final step constructs the final t-facet tree to be displayed to the user.

In the first step, this method collects a set of features for each user and query aiming to capture the deemed relevance of a given t-facet. Three groups of features are computed for each t-facet, query and user tuple. ***Group-CF*** contains collaborative filtering features, ***Group-P*** contains personalization features and ***Group-Q*** includes features reflecting the query relevance. These three groups are then fed into a DNN trained to predict a t-facet ranking score. The higher the score, the higher the relevance of the t-facet to the user and the query.

Personalization Features (Group-P) This group of features incorporates individual user preferences into the ranking process. Previous user ratings are used to build a preference profile for each user. When the user rates a visited POI positively, it is assumed that the user also likes its categories. The same also applies for POIs rated as neutral or negative. Based on this assumption, we compute the following features:

- ***uf_positive_prob***: The probability that user rates the t-facet as positive in general. It is the number of t-facets positively rated by the user, divided by the number of all t-facets rated by the user.
- ***uf_positivity_rate***: The probability that the user rates this t-facet as positive when she/he sees it. It is different from the previous feature as it considers only the ratings given by the user to this specific t-facet.
  To obtain it, the number of times the user rated this t-facet as positive is divided by the total number of times the user rated this specific t-facet.

This features along with the previous one reflect whether the user has a strong attitude towards the t-facet on its own, or whether it depends on the individual document content.
– Similarly, features for the neutral t-facets (***uf_neutral_prob*** and ***uf_neutrality_rate***), and the negative t-facets (***uf_negative_prob*** and ***uf_negativity_rate***) are obtained.

In addition to these, other features characterizing the user are also added:

– ***user_gender***, ***user_age_group***. User age is mapped into fixed intervals (for example 10 to 15 years). Both gender and age reflect the user demographic interest in a specific t-facet.
– Additional features like ***user_avg_ratings***, ***user_reviews_count***, ***user_likes_count*** can also be added if they are available in the dataset.

The features are calculated for each individual user and t-facet pair. They can be pre-calculated offline for all the t-facet taxonomy and stored in the user profile in advance. This profile will be updated as user rates new POIs.

Collaborative Filtering Features (Group-CF) This group of features reflect the general searchers attitude towards the t-facets. Similarly to the user-based features, we compute the following list of collaborative-based features:

– ***cf_positive_prob***: The probability that users rate the t-facet as positive in general. It is the number of t-facets rated positively by all the users, divided by the number of all t-facets in the system.
– ***cf_positivity_rate***: The probability that the users rate this t-facet as positive when they see it. It is calculated as the number of times users rated this t-facet as positive divided by how many times the users rate this specific t-facet.
– In the same way, four features for neutral t-facets (***cf_neutral_prob*** and ***cf_neutrality_rate***), and negative t-facets(***cf_negative_prob*** and ***cf_negativity_rate***) are also obtained.

The features are calculated for each t-facet in the system, and can be pre-computed offline and stored in a global profile and updated periodically. Additional collaborative filtering features are extracted for each relevant POI and aggregated by averaging their values on the t-facet level, as listed below:

– ***avg_rating***: Average ratings for POIs associated to this t-facet.
– ***avg_rating_count***: Average number of ratings the POI received.
– ***avg_reviews_count***: Average number of reviews the POI received.
– ***avg_reviews_sent***: Average reviews polarity. In order to calculate this feature, sentiment analysis is performed for the most recent reviews of the POI. Then, the sentiments are averaged into one overall polarity score.
– ***price_group***: This feature rates the price group of this POI belong. Some users prefer cheaper places or more expensive ones which should be considered during the ranking process.

- ***avg_cat_count***: The category count (i.e. the different number of category types associated with this POI).
- ***avg_cat_depth***: It reflects the depth of the category in the hierarchy type tree. It considers the count of all categories regardless their level. This includes the parent categories as well. If the document belongs to more than one category, the sum of the depth of all categories is taken. Common ancestors are added only once.

Query Relevance Features (Group-Q) This group of features reflect the relevance of the t-facet to the input query and to the set of relevant results returned by the search engine in response to it. The features are:

- $\boldsymbol{max\_sim(f_i, q)}$: Maximum cosine similarity between t-facet name and each keyword in the query. This feature seeks to capture the direct mention of the category in the query.
- $\boldsymbol{avg\_sim(f_i, q)}$: Average semantic similarity between the t-facet and each keyword in the query. If multiple keywords are similar to the t-facet this will results in higher average and ultimately higher t-facet importance. The semantic similarities are computed using the cosine between the BERT vectors representing the two input texts.

Another set of feature are related to the t-facet information gain. It measures how much information is gained when the user selects this t-facet. This feature set employs the POI-level search engine score in several ways. It assumes that the relevance travels from the POIs to their t-facets:

- ***info_gain***: the average score of the POIs associated with this t-facet.
- ***mutual_info_gain***: similar to the previous one, but calculated only on POIs which are not covered by any earlier t-facets. A greedy approach is followed to calculate this feature. This feature highlights the importance of POIs which belong only to this specific t-facet.
- ***info_gain*@1**: It is the score of the top ranked POI seen by the user at first position, if they filter the results using this t-facet. This feature highlights t-facets associated with a POI with a high relevance score.
- ***info_gain*@*k***: Information gain at k is the average for the k top POIs seen by the user at the top of the result page, after they filter the results using this t-facet. k is the size of the search result page. The feature favors t-facets with many highly ranked POIs in the first page.
- $\boldsymbol{popularity(f_i, F_q)}$ is the t-facet popularity among the t-facets associated with the query results, computed as the number of relevant POIs that belong to this facet type divided by the total number of POIs in the result set.
- $\boldsymbol{mutual\_popularity(f_i, F_q)}$ is the t-facet mutual popularity among the t-facets associated with query results, computed as the number of unseen relevant POIs belonging to this t-facet divided by the total number of POIs in the result set.

Finally, a vector is computed with features from all the three groups. It includes also the t-facet and its ancestors names added as categorical features.

Features are extracted only for t-facets associated with relevant POIs returned by the search engine after the user submits the query.

DNN Model Architecture. In order to build the Deep Neural Network, each group of features is fed into a separate sub-network first, whose role is to reduce the features to a fixed number of nodes n per network. Regardless the different number of features in each group, the group's sub-network will encode it in exactly $n = 4$ nodes. The sub-networks are disconnected from each other, while their internal layers are fully connected. The sub-network outputs are then concatenated in a single layer used as input to the final prediction network. During training, the final network will utilize each sub-network output to predict the final t-facet relevance score.

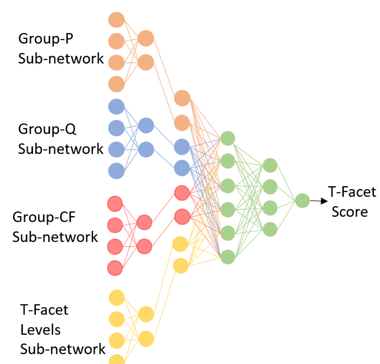

Fig. 1: The DNN architecture.

Figure 1 demonstrates the suggested architecture. The groups P, Q and CF features are first passed to the sub-networks. In addition the T-Facet and its ancestors names are encoded using one-hot encoding used as inputs to the yellow sub-network used to generate a dense vector, which acts as t-facet identifier during the training process. This architecture ensures an equal contribution from each feature group to the final scoring network, represented in green. It also avoids groups containing large number of features to dominate the training process. On the other end, the final network (green) determines the appropriate way to combine the group features into a single overall score. After generating the prediction score, a final step is needed to build the t-facets tree to be provided to the user. To build a final t-facet tree with v levels, we adopted a *Fixed Level (Max)* strategy [8]. The strategy uses a predefined fixed page size for each t-facet level. It starts by grouping t-facets at level-v by their parent. Then, it sorts the parent nodes at level-$(v-1)$ by the maximum score of their children, generated in step one (DNN Model), and so on up to level-1.

4 Evaluation

Datasets. Our approach is evaluated on a customized versions of TREC-CS 2016 dataset and Yelp Open Data dataset, following the personalised t-facet ranking methodology described in [6]. In TREC-CS, the t-facet taxonomy is derived from the Foursquare category hierarchy. The dataset has 27 users, 61 requests and an average of 208 t-facets per request to be ranked. The second dataset is customized from Yelp Open Dataset. It contains 1,456 users (requests) and average of 168 t-facets per request to be ranked. The taxonomy is derived from Yelp category taxonomy. Only the first two levels of both taxonomies are included. Query is formulated using tags collected from user previous ratings.

Evaluation Metrics. We follow the strategy used in Faceted Search task of INEX 2011 Data-Centric Track [12]. We report two metrics suggested by the organizers. The number of actions (#Actions) counts how many clicks the user has to perform on the ranked facets in order to reach the first relevant POIs in the first page of results. This metric is a proxy for the user's effort, which will help in answering our research question. We also include F-NDCG@9 metric. It reflects how many unique relevant POIs are covered by the first populated t-facet tree. The t-facet page size for both dataset was set to nine.

DNN Training Setup. The DNN model is implemented using Keras TensorFlow 2.4 Functional API [3]. The network is trained to learn the target t-facet rank computed as the number of relevant POIs belonging to this t-facet, collected from the POI level relevance judgment. The model uses Mean Square Error (MSE) as a loss function. All hidden layers in the model utilize the rectified linear activation function (ReLU). We used a pre-trained BERT model to compute the semantic similarity between the query and the t-facets. The ranking approach was evaluated using cross-validation on the request level. Learning rate is set to 0.0001, and the number of epochs is set to 200 for TREC-CS and 100 for Yelp.

Results and Discussion. Table 1 summarizes the results for the two datasets included in our experiments. At the top, we report the results of our DNN approach using all features (*Group-All*), each group individually (*Group-P*, *Group-CF*, and *Group-Q*), and their combinations (*Group-CF+Q*, *Group-CF+P*, and *Group-Q+P*). The second part of the table reports the results of personalized t-facet ranking baselines.

Table 1: Evaluation results using Fixed Levels (Max) tree building strategy.

Scoring Method	TREC-CS		Yelp	
	#Actions	F-NDCG@9	#Actions	F-NDCG@9
Group-All	1.368	**0.223**	**2.480**	0.058
Group-P	1.333	0.222	2.500	**0.059**
Group-CF	1.316	0.210	2.530	0.054
Group-Q	1.298	0.219	2.505	0.056
Group-P+CF	1.316	0.221	2.503	**0.059**
Group-Q+CF	1.333	0.214	2.536	0.054
Group-P+Q	1.579	0.179	2.512	**0.059**
Prob. Scoring [8]	1.333	0.141	2.982	0.034
VSM Scoring [7]	**1.281**	0.099	2.668	0.038
MF-SVM [4]	1.474	0.101	4.729	0.007
Most Prob. (Person) [9]	1.684	0.207	2.557	**0.059**
Most Prob. (Collab)[9]	1.386	0.204	2.506	0.053
TF.IDF [10]	1.316	0.157	3.223	0.035

From results, we can see that DNN-based models could effectively minimize the user effort on both datasets. Considering the number of actions metric, Group-All is the most performing system on the Yelp dataset, while Group-Q is the best system for the TREC-CS on which, however, Group-all has the extra benefit of better F-NDCG values. DNN-based models maximizes F-NDCG values across both datasets. They outperform all the baselines in TREC-CS dataset, and produce comparable values with the best two baselines in Yelp datasets. This reflects how the DNN-based models have the ability to generate trees with a collection of relevant t-facets, rather than producing a tree with

[3] Python implementation code for the DNN available at https://bit.ly/3AkCTGF

a single relevant t-facet at its top, as in the case of VSM Scoring method. Indeed, VSM scoring minimized the # of Actions but performs poorly in terms of F-NDCG, which means that the tree provided in the first page contains more irrelevant facets. Considering the individual groups of features, the performance of the groups seems comparable, it is fair to assume that the groups contribute equally to the results of Group-All model. Considering Group-P, in TREC-CS dataset, each user profile has either 30 or 60 ratings. Such a small number of ratings impacts negatively on the performance of this group of features. Conversely, Yelp dataset contains more user preference samples and interaction data. This helps in building richer user profiles to aid the personalization process. This is evident also across all the baselines that make use of user profiles, i.e. Prob. Scoring, VSM Scoring and Most Prob. (Person).

5 Conclusions

In this work, we introduced a DNN-based approach for learning to rank personalized type-based facets. The approach extracts features that reflect the t-facet relevance from multiple perspectives. Personalization was achieved in two ways. At feature level, by computing a set of user-based features. At the dataset level, by training the DNN model on target t-facet ranks, which reflect the user interests. The results have shown that our method improves the ranking process, especially with rich users' profiles. In the future, we intend to develop property-based facet ranking methods to generate an integrated facet ranking framework.

References

1. Abel, F., Celik, I., Houben, G.J., Siehndel, P.: Leveraging the semantics of tweets for adaptive faceted search on twitter. The Semantic Web (2011)
2. Aliannejadi, M., Mele, I., Crestani, F.: A cross-platform collection for contextual suggestion. In: SIGIR. ACM (2017)
3. Bayomi, M., Lawless, S.: Adapt_tcd: An ontology-based context aware approach for contextual suggestion. In: TREC (2016)
4. Chantamunee, S., Wong, K.W., Fung, C.C.: Collaborative filtering for personalised facet selection. In: IAIT (2018)
5. Chantamunee, S., Wong, K.W., Fung, C.C.: Deep autoencoder on personalized facet selection. In: Neural Information Processing (2019)
6. Esraa Ali, Annalina Caputo, S.L., Conlan, O.: Dataset creation framework for personalized type-based facet ranking tasks evaluation. In: CLEF 2021
7. Esraa Ali, Annalina Caputo, S.L., Conlan, O.: Personalizing type-based facet ranking using bert embeddings. In: SEMANTiCS 2021
8. Esraa Ali, Annalina Caputo, S.L., Conlan, O.: A probabilistic approach to personalize type-based facet ranking for poi suggestion. In: ICWE 2021
9. Koren, J., Zhang, Y., Liu, X.: Personalized interactive faceted search. In: WWW. ACM (2008)
10. Le, T., Vo, B., Duong, T.H.: Personalized facets for semantic search using linked open data with social networks. In: IBICA (2012)
11. Tvarožek, M., Bieliková, M.: Factic: personalized exploratory search in the semantic web. In: ICWE. Springer (2010)
12. Wang, Q., Ramírez, G., Marx, M., Theobald, M., Kamps, J.: Overview of the INEX 2011 Data-Centric Track. In: Focused Retrieval of Content and Structure. Springer (2012)