# A Context Based Recommender System through Collaborative Filtering and Word Embedding Techniques

Mahsa Khorasani, Hamid Sadjadi, Faezeh Ramazani, Faezeh Ensan*
Ferdowsi University of Mashhad
*ensan@um.ac.ir

## 1 Introduction

This report presents a description of the context-based recommender system that was developed by the FUM-IR team from the Ferdowsi University of Mashhad for the Contextual Suggestion track of TREC 2016. This will also include the description of the different runs were submitted to this track. In developing our system, we followed two main approaches for finding suitable attractions for a given user: a content-based approach and a category-based approach.

In the content-based approach, all Web pages related to attractions are modeled as vectors of real numbers using word embedding and document embedding techniques [1]. Then, similarities between attractions in the profile of a given user and new attractions are calculated using methods for finding similarities between vectors. In the category-based method, a subset of attractions is modeled as a vector of categories. These categories are extracted from the category information of the related Yelp, TripAdvisor, or Foursquare pages of the attractions. In addition, a user profile is modeled as a vector of categories, where these are categories are extracted based on a mapping from the tags provided in the user's profile and the categories extracted for the attractions. Finally, similarities between attractions and user profiles are calculated based on similarities between these vectors. We submitted three methods of combining these two approaches to this track as three different runs.

In the following, we will describe our system thoroughly and explain the different phases of its development.

## 2 Our approach

The development of our systems underwent the following phases:

1. Information gathering and preprocessing
2. Content-Based modeling: Developing a word-embedding model for Web pages crawled for attractions
3. Category-Based modeling: Developing a category-based vector model for user profiles and attractions

4. Ranking and recommendation: Applying models developed in Phases 2 and 3 for finding and ranking related attractions to the user profiles

In the following, we will describe each phase in more detail:

## 2.1 Information gathering and preprocessing

In this phase, we processed the web crawls provided by the track chairs in order to make it appropriate for the word-embedding techniques. We needed to extract the main content of a page as parts of the corpus and discard irrelevant content such as HTML tags, commercial ads, and also the network connection error messages. For this purpose, we used the technology proposed in [2] and the code provided in [3] for extracting the main text content of all Web pages.

We also needed the category information of attractions. We processed crawls from TripAdvisor, Foursquare, and Yelp and extracted the information related to their categories and their average ratings. We created a mapping list from categories of Foursquare to Yelp and TripAdvisor to Yelp, manually. We used these mapping lists for modeling Foursquare, Yelp, and TripAdvisor attractions as vectors of Yelp categories. We also manually created a mapping list from user tags onto Yelp categories.

## 2.2 Content-Based Model

The main idea behind this model is to use the content of Web pages for finding their similarities. For example, for an Italian restaurant with Italian menu and romantic atmosphere, a coffee shop with tea, snacks and Italian foods that also offers classic music can be considered a related attraction. We used the codes provided in [4] for creating vectors for documents and finding similarities between each pair of document vectors.

## 2.3 Category-Based Model

In this phase, a set of vectors is developed for those attractions that have Yelp, foursquare or TripAdvisor pages, where each cell in the attraction vectors is associated to a category (or sub category) in Yelp. When an attraction belongs to a category, its associated cell on that vector is assigned to 1, otherwise it assigned to 0. Similarly, category vectors are made for user profiles, with the difference that a cell in the vector can have three values: 1 for categories of attractions that user liked them, -1 for categories of attractions that user did not like, and 0 for categories that have not been mentioned in the user profile, or are both in her liked and disliked list of attractions.

## 2.4 Ranking and Recommendation

In this phase, we applied models developed in Phases 2 and 3, and produced three set of results as follows:

### 2.4.1 Run #1

In this run, the category-based vectors are used for finding similarities between user profiles and attractions. We first filter all attractions with a rating less than the average (e.g. those that have ratings less than 2.5 out 5 in TripAdvisor). Then, we used cosine-similarities between vectors to find the most similar attraction to user profiles

### 2.5 Run #2

In this run, the document-embedding vectors and the similarities between them are employed to produce a list of the most similar attractions to each attraction in the user profile. We found that despite a lot of very related results, this list contains a couple of completely unrelated pages. Hence, we decided to filter the result set for having a more precise list of attractions. We made an intersection between these lists with the attractions provided in the first run, making them more precise in the cost of decreasing recall. Then, we unioned all attractions related to each liked attraction in the user profile in a set, sorting them based on their ratings on their source Web page (like TripAdvisor, and Foursquare).

### 2.6 Run #3

This run is very similar to the second run, with this difference that we employed a different method for sorting attractions. For each liked attraction in the user profile, we created a list in the same way we created it in Run #2. Afterwards, we iteratively selected two top attractions from each list and merged them to the final result set. We continue our iterations until we find 50 results from these lists.

## 3 Concluding Remarks

In this report, we explained the main phases of our process for creating context-based recommendations. For future work, we aim at analyzing the performance of content-based approaches with different filtering strategies to increase recall. We also aim at investigating and applying different methods for combining content-based and category-based approaches.

## 4 References


1. Mikolov, Tomas, et al. "Distributed representations of words and phrases and their compositionality." *Advances in neural information processing systems*. 2013.
2. Kohlschütter, Christian, Peter Fankhauser, and Wolfgang Nejdl. "Boilerplate detection using shallow text features." *Proceedings of the third ACM international conference on Web search and data mining*. ACM, 2010.
3. https://github.com/kohlschutter/boilerpipe
4. https://github.com/medallia/Word2VecJava