CIRGIRDISCO at TREC 2013 Contextual Suggestion Track: Using the Wikipedia Graph Structure for Item-to-Item Recommendation

M. Atif Qureshi[1,2], Arjumand Younus[1,2], Colm O'Riordan[1], and Gabriella Pasi[2]

[1] Computational Intelligence Research Group, National University of Ireland Galway, Ireland

[2] Information Retrieval Lab,Informatics, Systems and Communication, University of Milan Bicocca, Milan, Italy

muhammad.qureshi@nuigalway.ie,arjumand.younus@nuigalway.ie,colm.oriordan@nuigalway.ie, pasi@disco.unimib.it

Abstract. This paper describes our participation in the TREC 2013 contextual suggestion task. We fetch possible locations based on given contexts using Google Places API and WikiTravel. This is followed by a Wikipedia-based item-to-item similarity computation framework which uses the Wikipedia category-article structure to compute similarity between example locations rated by users and the suggested locations. This is then used in an item-based nearest neighbor recommendation framework to recommend the locations based on given user profile ratings.

1 Introduction and Task Description

According to a report from the The Second Strategic Workshop on Information Retrieval in Lorne (submitted to SIGIR Forum, 2012),"Future information retrieval systems must anticipate to user needs and respond with information appropriate to the current context without the user having to enter an explicit query" [1]. This led to the organization of a new TREC task within the information retrieval community: contextual suggestion track which is basically a task of "personalized location recommendation."

As input to the task, the participating groups were provided with a set of 562 profiles, 50 examples of tourist recommendations along with user ratings and 50 geographical contexts in CSV/JSON format. The tourist recommendation examples consisted of a title and a short description of a location as well as an associated URL, the organizers used Philadelphia, PA as the seed city and thereby the example recommendations were for locations in Philadelphia, PA. Each profile across a user comprised a list of rated URLs of the location recommendations with ratings divided into a rating for the title and description and a rating for the website of the location. Each context consisted of spatial information (city-name, state-name, latitude, longitude).

The task for the participating teams was to build a contextual suggestion system that automatically provides a ranked list of 50 suggestions corresponding to each profile and context pair. Each suggestion should contain a title, description and associated URL for each profile/context pair. This is the second year the TREC contextual suggestion task was organized and this year's task differed from the previous year's task in that this year's submissions allowed generated location suggestions to be either from the Open Web[3] or from the ClueWeb 2012 dataset[4] whereas last year's submissions were restricted to the Open Web only.

Our participation in the task comprises of an item-based recommendation framework where we compute similarities between items through the Wikipedia graph structure. The method comprises the following steps

1. Generating a set of suggestions using Google Places API and WikiTravel.
2. Using Wikipedia category-article structure for generation of similarity scores between the suggestions and examples.
3. Using an item-to-item recommendation algorithm using the similarity scores computed in the previous step.

2 Fetching Contextual Suggestions from the Open Web

The first step was to collect potential location recommendations across the different contexts. We did this in two steps as follows: 1) using the Google Places API with city and state of the location as query we retrieved a list of locations of various types. These types were based on the list of place types accepted by the Google Places API, and some examples include place types such as amusement parks, art galleries, museums, aquariums, bars, book stores, bowling alleys, cafes, casinos, movie theaters, parks etc., and 2) using the WikiTravel pages of the given locations (i.e. city and state) we extracted the "To See" section containing a list of tourist attractions.

Short descriptions corresponding to locations were obtained using Wikipedia extended abstracts for location titles that matched the Wikipedia article titles, and using snippets obtained from Bing search API for cases where the Wikipedia article title matching does not occur. We submit two runs to the TREC 2013 contextual suggestion task where the first run assigns equal priority to locations fetched from both Google Places and WikiTravel while the second run assigns a higher priority to locations fetched from WikiTravel.

3 Computation of Item-to-Item Similarity using Wikipedia Category-Article Structure

The steps defined in section 2 enable us to create a number of suggested locations for each context. From these locations we generate recommendations corresponding to each profile by computing a similarity score between the rated examples

[3] This implies usage of external tourist websites and APIs such as Google Places, Foursquare, Yelp, WikiTravel etc.

[4] http://lemurproject.org/clueweb12/index.php

within each profile and the fetched suggestions. In this section we present our strategy to exploit the Wikipedia category-article structure for computing similarity between example locations and suggested locations that were fetched using Google Places API and WikiTravel as described in section 2.

First we extract possible n-grams from an example's description, and then we score relatedness of these n-grams with a location suggestion. The extracted n-grams are reduced to those that match the title of a Wikipedia article. We then fetch all the parent categories and all sub-categories to a depth count of two of the reduced n-grams' Wikipedia article. We refer to these categories as RC. Next, we retrieve the set of all articles within the Wikipedia category set RC (we refer this set as $Articles_{RC}$). Finally, all categories associated with these articles are retrieved which we refer to as WC; note[5] that RC is a subset of WC.

As the next step, we extract n-grams from the suggested locations' descriptions applying a similar reduction process of retaining the n-grams that match the title of a Wikipedia article. The reduced n-grams from suggested locations' descriptions which are contained in $Articles_{RC}$ are called matched phrases. We use these matched phrases to calculate the relatedness score. The following summarizes important factors which contribute in calculating our relatedness score for a location using Wikipedia category-article structure.

- $Depth_{significance}$ denotes the significance of category depth at which a matched phrase occurs; the deeper the match occurs in the taxonomy the less its similarity to an example location.

$$Depth_{significance}(p) = \sum_{cat \in RC \cap p_{categories}} \frac{1}{depth_{cat} + 1}$$

- $Cat_{significance}$ denotes the significance of a matched phrase's categories corresponding to the example location. The more categories of matched phrase in RC, the higher the similarity.

$$Cat_{significance}(p) = \frac{|RC \cap p_{categories}|}{|WC \cap p_{categories}|} * log(|RC \cap p_{categories}| + 1)$$

- $Phrase_{significance}$ is a combination of phrase word length and frequency of the phrase within the suggested location's description. The greater the word length of a phrase, the more informative or important it becomes, likewise the more frequent the phrase is in the suggested location's description, the more important the phrase is.

$$Phrase_{significance}(p) = log(wordlen(p) + 1) * p_{frequency}$$

Finally the equation for calculating the similarity score for locations based on Wikipedia category-article structure is:

[5] E.g., Wikipedia article "Steve Jobs" of "Apple Inc." contains a category "1955 births" which is not present either in parent nor in sub-categories of entity's Wikipedia article.

$$= \sum_{p \in MatchedPhrases} Depth_{significance}(p) \times Cat_{significance}(p) \times Phrase_{significance}(p)$$

The data for Wikipedia articles' hyperlinks, and Wikipedia category-article structure is obtained through a custom Wikipedia API that has pre-indexed Wikipedia data and hence, it is computationally fast[6]. The API is developed using the DBPedia [2] 2012 dumps.

We then use the following equation to make predictions based on user profile ratings corresponding to the example locations[7].

$$prediction(suggested_{location}, example_{location}) = \frac{\sum_{l \in ratedExamples} similarity(suggested_{location}, example_{location}) * rating_{example_{location}}}{\sum_{l \in ratedExamples} similarity(suggested_{location}, example_{location})} \tag{1}$$

4 Conclusions

As a summary, our runs are based on an item-item based similarity metric that is normally used in collaborative filtering. However, unlike traditional collaborative filtering approaches we make use of Wikipedia category graph and Wikipedia article graph to compute similarity between places fetched from Google Places and WikiTravel. The descriptions of example suggestions given as part of user profiles are decomposed into n-grams and from within these n-grams we filter those which have a corresponding Wikipedia entry (i.e., a Wikipedia article); finally we determine an intersection between these n-grams and the Wikipedia article titles extracted from n-grams of returned places' descriptions (using Google Places API and Bing API). The computed intersections (precisely, Wikipedia articles) are used in a score computation framework that indicates a measure of similarity between example suggestions and returned places.

References

1. J. Allan, B. Croft, A. Moffat, and M. Sanderson. Frontiers, challenges, and opportunities for information retrieval: Report from swirl 2012 the second strategic workshop on information retrieval in lorne. *SIGIR Forum*, 46(1):2–32, May 2012.
2. C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. Dbpedia - a crystallization point for the web of data. *Web Semant.*, 7(3):154–165, Sept. 2009.

[6] http://www3.it.nuigalway.ie/cirg/prj/WikiMadeEasy.html, we aim to release the API as an open source Wikipedia tool to facilitate other researchers.

[7] Note that this form of ratings prediction is used in an item-based nearest neighbor recommendation framework