Finding Tours for a Set of Interests

Mohamed Abdel Maksoud*
Codoma.tech Advanced Technologies
Egypt
mohamed@amaksoud.com

Gaurav Pandey*
University of Jyvaskyla
Finland
gaurav.g.pandey@jyu.fi

Shuaiqiang Wang
Data Science Lab, jd.com
China
wangshuaiqiang1@jd.com

ABSTRACT

This paper addresses a novel tour discovery problem in the domain of travel search. We create a ranking of *tours* for a *set of travel interests*, where a tour is a group of city documents and a travel interest is a query. While generating and ranking tours, it is aimed that each interest (from the interest set) is satisfied by at least one city in a tour and the distance traveled to cover the tour is not too large. Firstly, we generate tours for the interest set, by utilizing the available ranking of cities for the individual interests and the distances between the cities. Then, in absence of existing methods directly related to our problem, we devise our novel techniques to calculate ranking scores for the tours and present a comparison of these techniques in our results. We demonstrate our web application *Travición*, that utilizes the best tour scoring technique.

CCS CONCEPTS

• **Information systems → Retrieval tasks and goals**;

KEYWORDS

Information Retrieval, Ranking, Search Result Organization

ACM Reference Format:
Mohamed Abdel Maksoud, Gaurav Pandey, and Shuaiqiang Wang. 2018. Finding Tours for a Set of Interests. In *WWW '18 Companion: The 2018 Web Conference Companion, April 23–27, 2018, Lyon, France.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3184558.3186982

1 INTRODUCTION

In this paper, we present our novel method to address the problem of finding *tours* (or groups of cities) in response to a *set of travel interests*. When planning a trip, it is likely that a user desires her multiple travel interests to be catered. While it can be difficult to find many individual cities that satisfy a set of interests, the creation of relevant and practical tours is expected to enrich the offering to the user. We create ranking of tours in response to a set of interests, where each tour is a set of city documents and the set of interests is a combination of travel related queries (interests). To elaborate, for a set of interests $\{itr_1 \dots itr_n\}$ we create a ranking of tours $\langle t_1, \dots t_m \rangle$, where each t_i contains one or more cities. For this, we utilize the already available rankings of city documents for individual interests to form tours, while aiming that each interest in the interest set is satisfied by at least one city in the tour.

We see that extensive research has been carried out to cluster searched documents, in order to increase the coverage of results presented to the user [3, 4]. Also, there are many efforts for geo-spatial routing between locations [9, 14], rank aggregation [5] and group recommendation [12]. Moreover, there are recent efforts aimed to provide tours to users [2, 6, 7, 10]. To the best of our knowledge, there are no existing methods that address the tour creation problem in our way: i.e. a) by utilizing document rankings for individual queries, to generate *document groups* and rank them in response to a particular *set of queries*, b) while aiming that a top ranked document group satisfies all the queries in the set.

Our method initially generates the candidate tours and then uses novel scoring techniques to rank them. We utilize city rankings using our *CitySearcher* algorithm [1], that ranks cities (i.e. documents with information about one city each), in response to a travel interest (query). In response to a set of interests, we prune the city rankings for individual interests, to keep only the highly relevant cities. Then, we form the collection of candidate tours, so that each tour includes 0 or 1 city from each pruned city ranking for interests. Hence, for n interests, the candidate tours would be sets of cities of size 1 to n. We further filter out the candidate tours in which the cities are too far apart, since such tours are not practical. Since our method is unique and does not have related existing baselines, we rank the candidate tours using our novel and relatively simple scoring techniques, that use different combinations of a) available city-interest relevance scores and b) distance required to cover cities in a tour. On experimentation, the best scoring technique is identified and used in the *Travición* web-application: www.travicion.com, that we demonstrate.

Our presented web-application is intuitive and has an easy-to-use interface, that enables the user to select her set of travel interests and finds relevant tours for her.

2 METHOD

Here we describe objectives of *Travición* web-application, followed by tour generation, tour scoring techniques and implementation.

2.1 Objectives

Given: We have the following information available:

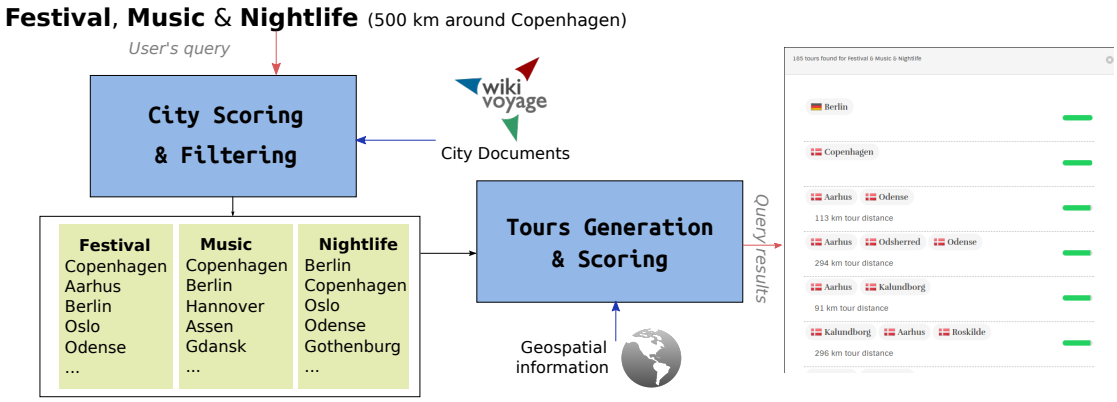
- A set of m documents representing different cities $\{c_1 \dots c_m\}$, having one to one mapping between cities and documents.
- A set of n travel interests $\{itr_1 \dots itr_n\}$ (provided by user), where each itr_i acts as query on city documents.
- A scoring function $score(c, itr)$ that gives a relevance score to a city c for an interest itr. This allows us to rank the m

*Indicates equal contribution

WWW '18 Companion, April 23–27, 2018, Lyon, France

ACM ISBN 978-1-4503-5640-4/18/04.
https://doi.org/10.1145/3184558.3186982

Figure 1: Architecture of *Travición*



cities for each interest, i.e. for the interests $\{itr_1 \dots itr_n\}$ we have the corresponding city rankings: $\{cr_1 \dots cr_n\}$.

- A function $dist(C)$ that calculates the shortest geographical distance to travel cities $C \subset \{c_1 \dots c_m\}$, while starting and ending at any city in C.

Aim: For the set of interests $\{itr_1 \dots itr_n\}$, generate a single ranked list of tours i.e. $\langle t_1, \dots t_m \rangle$, where each tour t_j is a set of cities and $1 \leq |t_j| \leq n$.

Requirements: The ranking of tours should satisfy the following:

- Higher rankings should be assigned to those tours which address each interest, i.e. the tour contains cities, such that each interest is addressed by at least one city.
- Higher rankings should be assigned to those tours which contain cities that are not too distant from each other.

Remarks: It should be noted that we limit our scope and do not let the number of cities in a tour to be more than the number of interests, since one city is sufficient to address an interest. We admit that this would exclude some desirable tours containing two or more cities in close proximity and relevant to a particular interest.

2.2 Tour Generation

Given m cities and an an interest set with n interests, in order to create tours with 1 to n cities, there would be $\binom{m}{n} + \binom{m}{n-1} \dots \binom{m}{1}$ candidates, which could be a very large number. Therefore, for the n interests we generate the city rankings i.e. $\{cr_1 \dots cr_n\}$ (m cities each) [1], and prune them to keep only top k cities ($k << m$), as the cities occurring low in the city rankings of all the interests are quite unlikely to make good tours. Therefore, forming tours of size 1 to n, by taking 0 or 1 city from each of the n pruned city rankings with k cities each, there would be maximum $(k+1)^n - 1$ candidates. The set of candidate tours to be ranked can be denoted as $T = \{t_1 \dots t_p\}$, where $p \leq (k+1)^n - 1$ is the number of tours and each tour t_i is also a set of cities such that $0 < |t_i| \leq n$.

Moreover, as it can be expected that many of such tours would contain cities that are too far apart, we also put a threshold of D km on the pair-wise distance between cities in the same tour. Therefore, we filter out all such tours t_i from T where the distance between any pair of cities is greater than D. Note that we use pair-wise distance rather than the whole distance covered by the tour for performance reasons: organizing cities in a k-d tree [11] allows for efficient proximity search given a certain radius. This geospatial filtering further reduces the number of tours in T considerably.

2.3 Tour Scoring Techniques

We propose the following techniques to score a candidate tour t ($t \in T$) for the combined set of interests $I = \{itr_1 \dots itr_n\}$:

2.3.1 Distance-wise Scoring. A naïve approach could be to score a tour t as inverse of the distance covered in it i.e. $dist(t)$:

$$distScore(t, I) = \frac{1}{dist(t)} \tag{1}$$

The above scoring function would completely favor tours containing less cities, and is expected to rank single city tours at the top. Although the candidate tours consist of cities ranking high for interests, the function itself is independent of any relevance for interests.

2.3.2 Average Relevance Scoring. Since an interest is satisfied by even one city in a tour, we define the tour-interest relevance as the maximum of city-interest relevances for the cities in the tour, i.e.:

$$rel(t, itr) = max(score(ct_1, itr), \dots score(ct_x, itr)), \tag{2}$$

where $ct_1, \dots ct_x$ are the x cities contained in tour t. Using this, we calculate the score of a tour for an interest set $I = \{itr_1 \dots itr_n\}$, by calculating the average of relevances for individual interests as:

$$relScore_{avg}(t, I) = \frac{\sum_{i=1}^{n} rel(t, itr_i)}{n} \tag{3}$$

2.3.3 MaxMin Relevance Scoring. This technique is based on our intuition that it is important for a tour to have high relevance for *each* interest in the interest set I. If a tour is highly relevant for all interests in I except one, it would not satisfy the set I in entirety. To emphasize the impact of the interest to which the relevance of tour is the least, we define the following score:

$$\begin{aligned} relScore_{mm}(t, I) = max(rel(t, itr_1), \dots rel(t, itr_n)) \\ \times min(rel(t, itr_1), \dots rel(t, itr_n)) \end{aligned} \tag{4}$$

2.3.4 Hybrid Scoring. We see that Distance-wise scoring considers only the tour distance and ignores the relevance. On the other hand, Average Relevance and MaxMin Relevance scoring techniques, do not take into account the tour distances. Therefore, we combine them to derive two hybrid techniques, expecting that a combination would lead to improvement in tour ranking.

The first hybrid scoring combines Distance-wise Scoring and Average Relevance Scoring by combining the scores calculated in Equations 1 and 3:

$$hybScore_1(t,I) = \lambda_1\, distScore(t,I) + (1-\lambda_1)\, relScore_{avg}(t,I) \quad (5)$$

Similarly, second hybrid scoring combines Distance-wise Scoring and MaxMin Relevance Scoring by combining the scores calculated in Equations 1 and 4:

$$hybScore_2(t,I) = \lambda_2\, distScore(t,I) + (1-\lambda_2)\, relScore_{mm}(t,I), \quad (6)$$

where λ_1 and λ_2 are the weighting parameters used in combining the scores.

2.4 Implementation

Travición is implemented as a web-application and takes a set of interests as input from the user. It utilizes the English version of Wikivoyage[1] dataset containing 6691 documents, where predominantly each document is representative of a particular city. For deriving ranking of cities for individual interests along with the ranking scores, it uses CitySearcher approach [1].

In the pruning step as described in Section 2.2, we limit the city ranking lists to their top 1000 results (i.e $k = 1000$). Distance between two cities is calculated by determining their respective latitude and longitude using the GeoLite dataset[2] and then using the Haversine formula [13]. Moreover, the threshold of pair-wise distances in one tour is set to 200 km (i.e. $D = 200$ in Section 2.2), so that the tours with one or more pair-wise distances more than D are not considered as candidates. Moreover, since in our evaluation (Section 4) $relScore_{mm}$ is the best performing technique, *Travición* employs it to score the tours. Additionally, the implementation also includes the functionality to exclude tours that are outside a specific region, i.e. the tours that lie outside a particular radius from a central location (this location and radius are optional inputs from the user). Figure 1 shows the architecture of *Travición* with the help of an example.

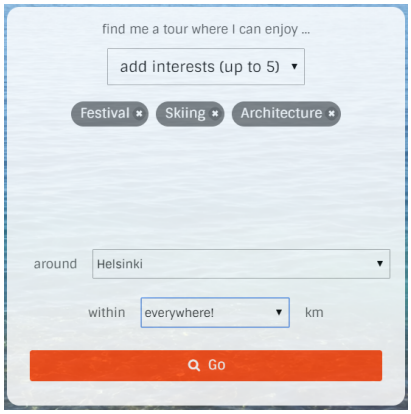
3 DEMONSTRATION SCENARIO

We plan to demonstrate our web application to the attendees, in which they can enter their travel interests to find a ranking of tours. Figure 2 shows the user interface of the *Travición* web application, which is quite intuitive. The user can choose up to 5 travel interests and press the 'Go' button. Thereafter, a ranked list of tours is presented to the user.

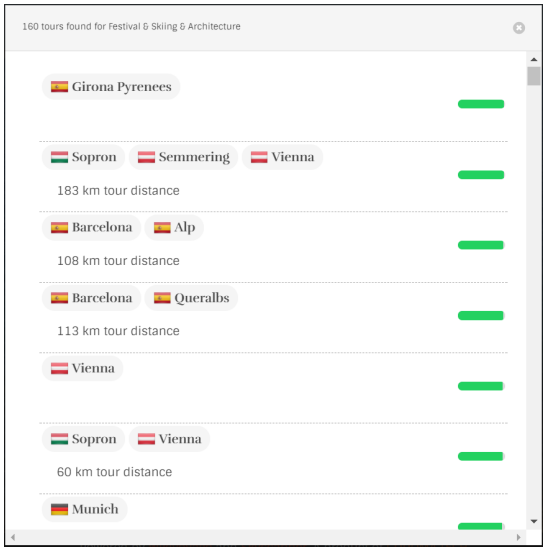
The search interface also allows the user to specify a specific region where they seek to do the tour. This region is specified by a center point (labeled 'around') and a radius in kilometers (labeled 'within'). Specifying a region effectively restricts the cities examined to those which are located in that region. A typical use

[1]https://en.wikivoyage.org/wiki/Main_Page
[2]https://dev.maxmind.com/geoip/geoip2/geolite2/

Figure 2: *Travición* User Interface



Tour Search



Search Results

case is when a user has a general idea where they want to travel (e.g. West Europe, East Asia), but they don't have specific destinations in mind. Therefore, she could choose a city in the mid of that region and specify the radius. Alternatively, if she wants to find tours around her current location, she shall choose it as the center point. Choosing 'everywhere' as radius would effectively consider all the world cities for tour formation. Hence, we demonstrate the scenario where a user specifies travel interests (and optionally a specific region) to discover a ranking of tours. The user shall expect that a high ranked tour would satisfy all her expressed interests (i.e. each interest should be addressed by at least one city in tour).

The ranking of tours contains single-city as well as multi-city tours as results. It should be noted that if the user selects only one interest, she would be presented with only single city tours. Moreover, if the user uses the 'everywhere!' option for the maximum distance from initial location, there are typically less multi-city tours in the top positions. This is because, with no restriction on

distance from initial location, many big cities around the world that can cater to multiple interests would rank high as tours.

On clicking a particular tour, the user is shown the tour on a map and then is also enabled to search for flights from an airport she specifies to one of the cities in the tour (not shown in the image). Flight and accommodation planning is not closely related to our demonstration or the problem we are presenting in this paper.
The web-application can be used on the web-page:
www.travicion.com
Also, an explanatory video / advertisement is available on:
www.youtube.com/watch?v=HlzmxSA2bjw

The hardware requirements for our demonstration are minimal, as the web-application only needs internet connection, and can be used on any modern web browser (on a computer or mobile device).

4 EXPERIMENTAL EVALUATION

For evaluation, we created a list of 50 travel interest sets, by choosing the most frequent travel interest sets requested on *Travición* website. Each interest set contains 3 or more interests and the average number of interests per interest set is 3.62. The experimental setup is same as in Section 2.4, except that we evaluate all the 5 scoring techniques ($distScore$, $relScore_{avg}$, $relScore_{mm}$, $hybScore_1$ and $hybScore_2$) described in Section 2.3. For this, we collect relevance assessments for top tours for these methods. We collected around 2500 ratings from 20 people using the crowdsourcing platform clickworker[3]. The ratings are taken on a scale of 1 to 20.

The standard ranking accuracy metric: normalized discounted cumulative gain (NDCG@1-5) [8] is used to compare our scoring techniques. Table 1 presents the experimental results for the scoring techniques: $distScore$, $relScore_{avg}$, $relScore_{mm}$, $hybScore_1$ and $hybScore_2$, formulated in Equations 1, 3, 4, 5 and 6 respectively. Weighting parameters λ_1 and λ_2 used in the scoring techniques $hybScore_1$ and $hybScore_2$ respectively, are assigned the value 0.5.

As expected, $distScore$ shows the worst results across the metrics, since it ranks the tours only on the basis of distance traveled within a tour. We see that $relScore_{mm}$ is the best performing technique and is significantly better than $relScore_{avg}$, confirming our intuition that the relevance of a tour to each interest in the interest set is more important than the average relevance for the interests.

Moreover, we see that the $hybScore_1$ (hybrid of $distScore$ and $relScore_{avg}$) shows slight improvement over $relScore_{avg}$. Also, $hybScore_2$ (hybrid of $distScore$ and $relScore_{mm}$) does not show any improvement over $relScore_{mm}$. Overall, the results of $relScore_{avg}$ and $relScore_{mm}$ do not differ a lot from their respective hybrids. Probably, this happens because the distances between the cities are already considered while filtering the candidate tours, the inclusion of tour distances in scoring is not very effective.

5 CONCLUSION AND FUTURE WORK

This paper demonstrates the *Travición* web-application that displays a ranking of tours in response to a set of travel interests. For this, we addressed the unique problem of creating and ranking document groups for a set of queries, so that each query is satisfied by at least one document in each group. We presented our novel techniques for ranking tours (groups of cities) for a set of travel interests, so that a tour satisfies all interests as well as cities in the tour are in geographical proximity. On comparison, we observe that the best results are achieved when a tour is scored using the product of its best and worst relevances for interests in the interest set.

In future, we plan to consider practical distances between cities within a tour, as we have used a rather naïve approach to calculate distances using Haversine formula. This is important especially when there are no direct roads between the cities, or there is a lake or mountain between them. Also, we aim to explore consideration of multiple cities per interest for tour formation.

[3]https://clickworker.com

Table 1: Comparison of Ranking Methods

NDCG	@1	@2	@3	@4	@5
distScore	0.6862	0.7012	0.6901	0.6964	0.6911
$relScore_{avg}$	0.7994	0.7958	0.7939	0.7951	0.7980
$relScore_{mm}$	**0.8662**	**0.8562**	**0.8526**	**0.8507**	**0.8472**
$hybScore_1$	0.7994	0.7958	0.7942	0.7954	0.7996
$hybScore_2$	0.8632	0.8532	0.8507	0.8494	0.8446

ACKNOWLEDGMENTS

This work was supported by Codoma.tech Advanced Technologies in the context of the *Travición* project (www.travicion.com) and the Academy of Finland (MineSocMed Grant No 268078).

REFERENCES

[1] Mohamed Abdel Maksoud, Gaurav Pandey, and Shuaiqiang Wang. 2017. CitySearcher: A City Search Engine For Interests. In *SIGIR*. 1141–1144.
[2] Flora Amato, Antonino Mazzeo, Vincenzo Moscato, and Antonio Picariello. 2014. Exploiting cloud technologies and context information for recommending touristic paths. In *Intelligent Distributed Computing VII*. Springer, 281–287.
[3] Claudio Carpineto, Stanislaw Osiński, Giovanni Romano, and Dawid Weiss. 2009. A Survey of Web Clustering Engines. *ACM Comput. Surv.* 41, 3 (2009), 17:1–17:38.
[4] Carlos Cobos, Henry Muñoz Collazos, Richar Urbano-Muñoz, Martha Mendoza, Elizabeth León, and Enrique Herrera-Viedma. 2014. Clustering of Web Search Results Based on the Cuckoo Search Algorithm and Balanced Bayesian Information Criterion. *Inf. Sci.* 281 (2014), 248–264.
[5] Cynthia Dwork, Ravi Kumar, Moni Naor, and D. Sivakumar. 2001. Rank Aggregation Methods for the Web. In *WWW*. 613–622.
[6] Laura Martinez Garcia, Jhonatan Montes Serna, and Valentina Codutti. 2017. 10 SIGNALS: A personalized city tours prototype. In *COLCOM*. 1–6.
[7] Damianos Gavalas, Michael Kenteris, Charalampos Konstantopoulos, and Grammati Pantziou. 2012. Web application for recommending personalised mobile tourist routes. *IET software* 6, 4 (2012), 313–322.
[8] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated Gain-based Evaluation of IR Techniques. *ACM Transactions on Information Systems* 20, 4 (2002), 422–446.
[9] I. Johnson, J. Henderson, C. Perry, J. Schöning, and B. Hecht. 2017. Beautiful...but at What Cost?: An Examination of Externalities in Geographic Vehicle Routing. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 2 (2017), 15:1–15:21.
[10] Kwan Hui Lim, Jeffrey Chan, Christopher Leckie, and Shanika Karunasekera. 2017. Personalized Trip Recommendation for Tourists based on User Interests, Points of Interest Visit Durations and Visit Recency. *Knowledge and Information Systems* (2017).
[11] Songrit Maneewongvatana and David M. Mount. 1999. Analysis of approximate nearest neighbor searching with clustered point sets. *CoRR* cs.CG/9901013 (1999).
[12] Judith Masthoff. 2015. Group Recommender Systems: Aggregation, Satisfaction and Group Attributes. In *Recommender Systems Handbook*. Springer, 743–776.
[13] Jovin J Mwemezi and Youfang Huang. 2011. Optimal Facility Location on Spherical Surfaces: Algorithm and Application. *New York Science Journal* 4, 7 (2011), 21–28.
[14] Jie Shao, Lars Kulik, Egemen Tanin, and Long Guo. 2014. Travel Distance Versus Navigation Complexity: A Study on Different Spatial Queries on Road Networks. In *CIKM*. 1791–1794.