Siena College's Institute of Artificial Intelligence TREC 2016 Contextual Suggestion Track

Tristan Canova, Daniel Carpenter, Kevin Danaher, and Neil Devine
Advisor: Dr. Darren Lim

September 2016

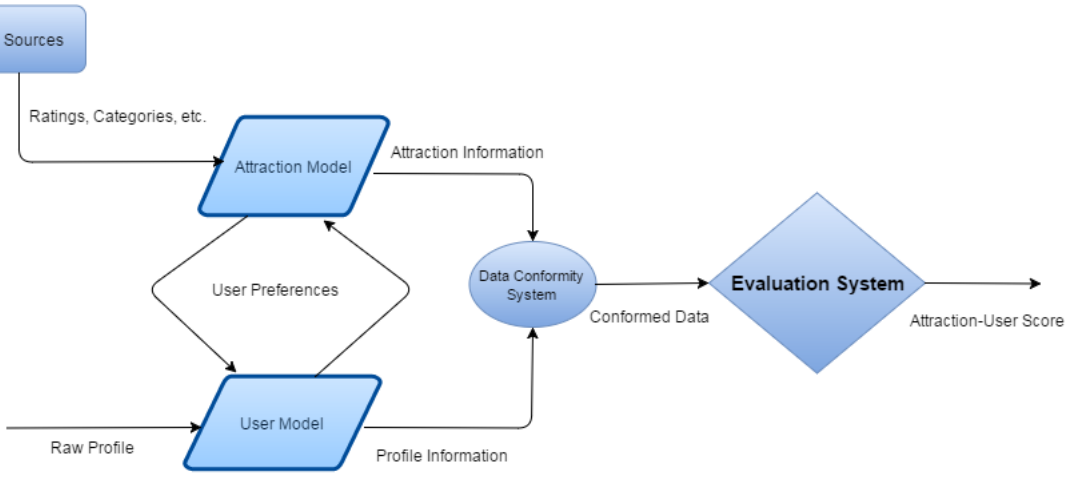
Abstract

This report contains an overview of Siena College's participation in the 2016 Contextual Suggestion track of the Text Retrieval Conference (TREC). The first task of our research was to devise a system to accurately and effectively gather relevant and complex information about specific points of interest (POI's) that were provided by TREC. The second task was to then take in user profiles and, based on the information in those profiles, develop a query-less search algorithm that would accurately predict the most relevant POI's in a given area (context). This list of POI's had to be ranked based on relevance for each given user profile.

To complete the task of gathering information on these POI's, our team utilized four API's: FourSquare, Google Places, Yellow pages and Trip Advisor. We also used Levenshtein distance in order to ensure accuracy when querying these API's. Once information on every point of interest was gathered, they were all ranked based on the user's perceived preferences, as well as a combined rating taken from users of the aforementioned sites.

1 Introduction

The Text Retrieval Conference (TREC) is a program that originated in 1992 and focuses on various forms of information retrieval, sponsored by the National Institute of Standards (NIST) and U.S. Department of Defense. TREC's purpose is to promote research in information retrieval by providing the infrastructure needed for big scale evaluation of information retrieval methodologies. Siena's Undergraduate Computational Contextual Evaluation and Suggestion System (SUCCESS) team participated in the TREC 2016 Contextual Suggestion track. Our goal was to create a system that returned a list of ranked suggestions of attractions in a particular location based on an individual's profile. Such a system could suggest several places of interest in a location foreign to an individual based on their favorite hobbies and places to go.



TREC provided our team with several contexts and user profiles. Contexts consisted of a context ID, city, state, latitude, and longitude. User profiles contained the individual's age and gender, the trip type and season, and a set of attractions that the user had previously rated from 0 (strongly uninterested) to 4 (strongly interested), or -1 if they didn't leave a rating. Additionally, each profile provided 30 attractions that our program was required to rank according to that user's interests.

Absorbing all of the given information, our SUCCESS team at Siena College designed an approach that compared information from four different API's: Google Places, Foursquare, Yellow Pages, and Trip Advisor. After a significant amount of data manipulation, we were able to suggest a ranked list of attractions for each profile-context pair.

2 Information Retrieval

2.1 Sources

To accurately model attractions, we collected information from numerous outside sources. We retrieved the most relevant attraction information from the following API sources:

- Google Places
- Foursquare
- Yellow Pages
- Trip Advisor

Each outside source provides a plethora of additional information on each attraction. Due to this abundance of information, we then gathered the most relevant and correct information for each attraction a user may visit.

2.2 Information Gathered

Each API source provides different attraction information in varying amount, quality, and relevance. In order to create an accurate model for a given attraction, we gathered the following information from each API source:

- Google Places
  - Rating
  - Reviews
  - Categories
- Foursquare
  - Rating
  - Reviews
  - Categories
- Yellow Pages
  - Rating
  - Reviews
  - Categories
- Trip Advisor
  - Rating
  - Number of Reviews
  - Categories
  - Certificate of Excellence
  - Seasonal Information
  - Group Information

With the aforementioned information, we modeled each attraction as a collection of information relevant to the majority of its visitor base. Such a model allows for subtle optimization of profile-attraction connections; thus, it is imperative that our information retrieval system is correct and accurate based on other sources of information.

2.3 Data Relevance

When given user generated (or automated) data, we do not assume that said data is both accurate and correct. Rather, we check each source of information with more accurate sources (TREC given data) to ensure that the data received is correct.

Despite having more accurate sources, it is important to model the difference in meaning between two potential query results. To handle nonconformity in data, we use a modified version of Levenshtein's Distance Algorithm to determine the relative equality of two query results.

3 Data Conformity

3.1 Overview

Information on a given attraction is taken from four separate sources, so many variations on attraction types are found. We discovered that reducing this number and enforcing conformity often leads to more relevant search results.

Our solution to this problem is complete data conformity in our attraction model. We implement this solution using Levenshtein's Distance Algorithm and predefined data categories to which our model conforms.

3.2 Levenshtein Distance

The Levenshtein Distance Algorithm is commonly used in BioInformatics to determine the similarity of different strings; however, the algorithm may also be used to determine the similarity in meaning between two strings based on their root words and sentence structure. We enforce data conformity in our attraction model by using the algorithm in the following manners:

1. Allowing only tags which are similar to our predefined categories to be considered relevant to a user.

2. Determining the relevance of API results to ensure that all data is correct.

3.3 Predefined Categories

When defining or modeling an attraction, tags or categories are largely influential. As our evaluation system is highly dependent on relevant attraction categories, we allow only a predefined set of possible categories to be considered. We then use a modified implementation of Levenshtein's Distance Algorithm to merge and conform data accurately when necessary.

By limiting the number of possible categories to a predefined set, the definition of an attraction becomes much clearer. Although these categories may be combined, the most popular, and therefore relevant, defining categories stand out.

This method of achieving data conformity gives consistent, static, data. Because this data is predictable, our evaluation system is more precise as a result.

4 Evaluation System

4.1 User Modeling

Now that every attraction was accurately modeled, this data could be used to model our users. To accomplish this, users had a score for each predefined category; a high score indicated that the user liked this category, while a low score indicated the opposite. These scores were generated based on the user's attraction ratings given in their preferences. If the user rated the attraction a 0, each category associated with it received a large deduction to its value. A rating of 4 constituted a great increase in those categories' values. Smaller adjustments were made for ratings of 1, 2, and 3, and if the user had not rated the attraction, no changes were made.

This model was then used to rank each user's candidates. For each category associated with a candidate, the user's score for that category was used to determine whether or not they would like that attraction.

4.2 Using Online Ratings

Online ratings that were gathered from the APIs were used in ranking attractions as well. Initially, if an attraction was rated highly across the four sources, that attraction would receive an increase in rank. Poorly rated attractions were not penalized. However, this method placed too much importance on ratings and wrongly boosted attractions that were popular, but not relevant to the user.

To ensure that online ratings would not skew an attraction's ranking, a good rating was only taken into account if that attraction represented a category that the user liked. The same was true for Trip Advisor's Certificate of Excellence.

4.3 Optimization

With relevant data, we were then able to optimize our overall search results by using several trends in the data. By having information on each attraction from both internal data and outside visitors, we could then distinguish which characteristics are most important to those who visit said attraction. We then scored each available candidate for each user profile based on this information.

After each candidate has been given a score, we determine the most relevant attractions merely by sorting the candidates by their overall score.

5 Results and Conclusion

5.1 Runs

Our team has submitted three runs to compete in the Contextual Suggestion track of 2016 each with a different focus. Our runs are as follows:

- CasualChocolate

  This run modeled users and attractions to properly identify relevant information on user-attraction pairs. Using this information, this

run scores each pair based on relevancy with an emphasis on common user preferences.

- SassyStrawberry

  This run was similar to our CasualChocolate run, but in addition rating user-attraction pairs, it also placed a higher emphasis on specific user preferences.

- VerbatimVanilla

  This run served as a control run; it merely returns the attractions in the order of the input.

5.2 Results

For the Contextual Suggestion track, the most important metrics are the NDCG@5 and the Precision@5 metrics. The results for our runs for these metrics are as follows:

Results		
Run ID	NDCG@5	P@5
CasualChocolate	0.2650	0.3828
SassyStrawberry	0.2543	0.3690
VerbatimVanilla	0.2119	0.3310

These results show that CasualChocolate and SassyStrawberry perform significantly better than our control run, VerbatimVanilla. The results also suggest that common user-preferences is a larger factor for both the NDCG@5 and the Precision@5 metrics, as CasualChocolate outperformed both other runs in these areas.