A Vertical Search Engine for Homestay Accommodation Hunting

Final Project for SI 650 Information Retrieval

Pai-Ju Chang, Yuan Li

ABSTRACT

Airbnb has provided people with much flexibility in searching homestay housing. However, with its limited searching and filtering options, Airbnb has rendered users some unnecessary inconvenience in seeking the appropriate accommodations. And we are hoping to tackle this problem in our project. We utilized Natural Language Processing (NLP) to pre-process user query, combined with a Named Entity Recognition (NER) approach. In such way, our search engine would allow users to enter whole-sentence query. Then we would be using Okapi BM25 as our main ranking algorithm, and to evaluate the results using Mean Average Precision (MAP).

1. Introduction

Founded in 2008, Airbnb, Inc has been known for its service provided to arrange or offer lodging, primarily homesteads, and tourism experiences. Airbnb has become especially popular with youngsters, and recently more and more older people are feeling engaged with this platform as well. In order to smooth the process of temporary accommodation searching, we are intended to build a domain-specific housing searching engine. This system would allow users to search by whole-sentence query, not limiting to just single-word query. In this sense, we believe it would provide more flexibility for users in terms of house searching. Such domain-specific searching system could also potentially be integrated to a voice-recognition system, that allows users to get the optimal housing choices with much convenience.

2. Problem Definition & Data

The search engine on Airbnb's platform merely allows users to input their preferred location for getting preliminary results (Figure.1). Afterward, users could further shape their queries by adjusting multiple types of filters, including the date picker and price range slider (Figure.2). Although users may obtain accurate search results after several tries playing with these filters, the search process is perplexing and not straightforward enough since it requires users to finish multiple steps for getting the refined results. Therefore, we believe creating a single full-text search engine that integrates all types of filters into a simple text input will improve users' experience of searching on Airbnb's platform.

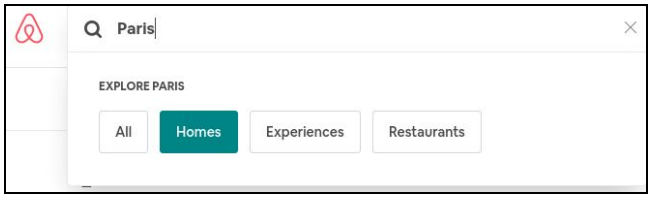

Figure 1: The search engine on Airbnb's platform. It only supports users to make a query by typing the name of the location.

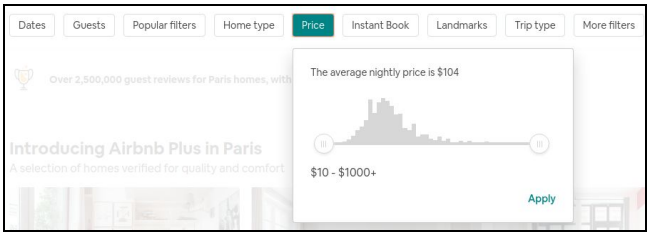

Figure 2: The filter tab on Airbnb's platform. Users are able to refine their searches and eliminate unrelated listings by adjusting those filters.

In addition to the incoherent search process, Airbnb's search engine does not support users to make a query based on textual descriptions of the listings. In other words, if terms of the query users use in searches are not encoded into a filter component, then they will not have any effect on the search result. We hope to propose a new search engine that allows users to customize their search more freely with no limitation on using filter components only.

Airbnb has strict restrictions on their API usage for protecting their data. Frequent attempts and accesses are not allowed and would cause the account banned from future services. Fortunately, a third-party website, Inside Airbnb [1], provides information of historical property listings on Airbnb, including URLs, textual descriptions of room's/house's condition, the description about the rule setting by hosts, the prices, locations of rooms/houses, and more. We eventually apply our system evaluation to the data accessed from Inside Airbnb. To reduce the difficulty of tasks, we only retrieve historical property listings in New

York, Los Angeles and Chicago, the top 3 most populated cities in the United States. Table 1 shows the number of listings for each city.

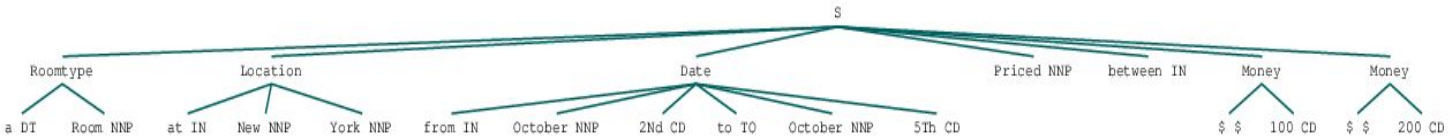

Figure 3: a linguistic parse tree for a simple query sentence (A room at New York from October 2nd to October 5th with price between $ 100-$200).

City	Number of listings
New York	95598
Los Angeles	94776
Chicago	17376

Table 1: The number of house/room listings in New York, Los Angeles and Chicago.

3. Related Works

Many previous words can be found in terms of constructing or improving the performance of a domain-specific search engine. Wei, Barnaghi and Bargiela [2] introduced a semantic approach called Information Retrieval In the Semantic web (IRIS) in term of building a semantic search engine. It used the annotation ontologies to integrate current information to information repository. Shilpy Sharma [3] introduced a Machine Learning approach in terms of building a domain specific search engine. This system uses Naive Bayes Classifier decide whether the document topic is relevant or not, and then the system would serve as an active learner to prioritized the ranking of documents. Manad, Bentounsi, Darmon [4] utilized Named Entity Recognition (NER) and regularization method to process and match unstructured HR data for big company. They proposed an efficient way to handle vast amount of integrated multi-source data.

4. Methodology

Due to the refrained structure of queries in a domain-specific search engine, it is feasible for us to combine all search features, including filters and types of inputs, into a simple text input enabling natural language processing (NLP) query. There is no double that NLP is hard to master and may cause several misjudgments due to the complexity and diversity of queries. We eventually narrow the domain for accommodation hunting only by extracting 4 constraints that are frequently used in accommodation searches, namely **the price, the range of date, the location and room type**. In the following sections, we will elaborate more on the technique we used for the extraction, ranking, and retrieval.

4.1 Natural Language Processing

We utilized NLTK Named Entity Recognition (NER) taggers to extract constraints from NLP query. we came up with several queries that might be used in such search engine and investigated the structure of queries for identifying structured annotations or patterns that could be further classified as 4 constraints. Figure 3 shows a linguistic parse tree for a simple sample query. By observing architecture of the tree, we were able to recognize some patterns for each constraint using part-of-speech of words. Semi-structured texts can hence be semantically annotated from those identified patterns.

For example, the pattern for extracting Room type from queries is listed below

$$Roomtype: \ \{(<DT>|<CD>)+<JJ>*<NNP.?>+\} \ (1)$$

It indicates a RoomType phrase should be formed when NLTK chunker find an determiner (DT) or number (CD), followed by an optional adjectives (JJ) and at least one noun term. The samples of semi-structured texts extracted by this rules are “a private room”, “2 bedroom”, “a house”.

However, we encountered some difficulties when we tried to expand the diversity and complexity of query sentence. We discovered that if we added terms using in Airbnb’s filter features (ex: with Laundry Facilities, pet allowed) into our query sentence (ex: looking for a private room with laundry facilities), some of those terms were mistakenly recognized as one of 4 constraints due to the similarity of the part-of-speech structure. To deal with those false alarms, we applied Regex Expression and Stanford NLP library to our system, making sure that those phrases identified by our NER rules were truly reflected those 4 constraints. Stanford NLP library is able to identify terms referring location, person or organization from plain texts. It thus was

used for confirming if semi-structure texts that were recognized as Location phrases contain information about the property's location. For Regex expression, we used it for eliminating terms that were not related by defining word pool for each type of phrases. For example, the nouns in Date phrase were restrained to those terms describing month (January, February...etc.) and their abbreviation (Jan, Feb….etc.)

4.2 Filtering and Ranking

After identifying phrases of constraints from the query sentence, we further filtered listings collected from inside Airbnb based on these phrases, making sure that the further process will only be applied to the listings fit to 4 constraints.

In order to rank the remaining listings based on their relevance with query sentence, we removed those phrases of constraints from the query sentence. Afterwards, for each listing, we obtain the descriptions of house, neighborhood, and facilities and applied Okapi BM25 ranking function on them. Based on the score, we are able to retrieve the top K documents which are most relevant to users' searches.

5. Results and Evaluations

We conducted a mixed approach in term of evaluation for this project. The two main sources for getting queries are Craigslists [5] and Facebook group [6]. We gathered 120 user posts in terms of seeking long-term and short-term accommodations, and used them as queries for system evaluation. Since there are too much complexity of user posts, we filtered out some of the posts with too much unrelated contents to accommodation seeking, and used those posts specifying housing criteria in particular. Figure 4 shows the result of the Average Precision (AP) values of each of the top 10 posts with most similar format as we used to build our system. The Mean Average Precision (MAP) value would be 0.59.

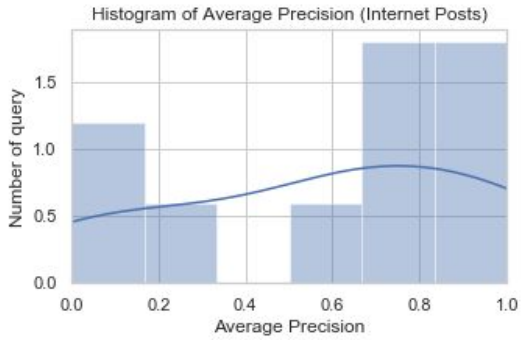

Figure 4: Histogram of Average Precision from online posts

Since user posts can only be approximated as housing queries, and furthermore, they may not necessarily be good samples for our housing queries. Meanwhile, we realized the descriptions posted on those platforms are not identical with those query sentence they put in the search engine. We then decided to recruit participants for generating queries from true users. For the two participants that we recruit, we asked each of them to come up with 10 queries -- words that they normally would use to search housing on Airbnb, but instead of single word query, we asked our participants to use whole sentences to search. Figure 5 shows the Average Precision values of each participant. Overall, the Mean Average Precision (MAP) is 0.75. The improved performance might due to the fact that Internet posts usually contain much complexity, and may not directly pertaining to house searching criteria. For example, one user's post seeking long-term or temporary accommodation may emphasize his/her own personality, as well as other self-introduction contents, and thus would render unnecessarily verbosity to our system evaluation.

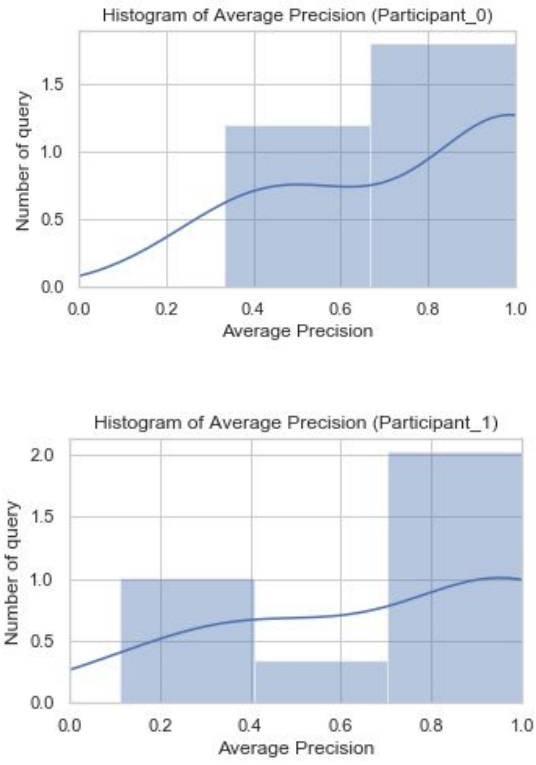

Figure 5: Histogram of Average Precision from 2 participants

6. Discussion

For this project, we have built a domain-specific housing search engine based on Airbnb, integrated with Named Entity Recognition method as well as other Natural Language Processing approaches. With such system, we

would be able to return the most relevant housing postings on Airbnb based on user's whole-sentence query. However, there are still some limitations of the system that we built.

First, it is extremely hard for our system to process complicated user query. As complexity grows in user query, our system performance would drop rapidly. One reason for the lack of flexibility of the system is that, we built the system upon single-word processing, which increases the possibility that words unnecessarily related would be recognized as a query word by our system, and use it to search through our database. This will also cause problem in handling negative term, for example, if a user specify housing to be "no pet", our system would recognize "pet" as a key word and traverse through database to return those results match exactly with "pet". This is a hard problem for our system. Due to time limitation of this course project, we do not have enough time to alleviate the problem; however, one solution that we have discussed is that, we can try using bigram or ngram to process user query, which would more precisely capture user needs.

Reference

[1] Inside Airbnb. Retrieved from: http://insideairbnb.com/

[2] Wei, Wang & Barnaghi, Payam & Bargiela, Andrzej. (2018). The Anatomy and Design of A Semantic Search Engine.

[3] Shilpy Sharma.Information Retrieval in Domain Specific Search Engine with Machine Learning Approaches.World Academy of Science, Engineering and Technology International Journal of Industrial and Manufacturing Engineering.Vol:2, No:6, 2008

[4] Manad, Otman & Bentounsi, Mehdi & Darmon, Patrice. (2018). Enhancing Talent Search by Integrating and Querying Big HR Data.

[5] los angeles apts/housing for rent - craigslist. Retrieved from: https://losangeles.craigslist.org/

[6] Holiday Accommodation Wanted and Advertised. Retrieved from: https://www.facebook.com/groups/Holiday.Accommodation.Wanted.Advertised/