Question Analysis in a Travel Domain Question Answering System

Sabita Acharya
University of Illinois At Chicago
851 S. Morgan
Chicago, IL, USA
sachar4@uic.edu

Amruta Nanavaty
University of Illinois At Chicago
851 S. Morgan
Chicago, IL, USA
abuch3@uic.edu

Abstract

Our main focus for this project is Question Analysis. We have proposed an approach to automatically assign labels to travel related questions based on the expected answer type.

1 Introduction

Vacations are a chance to take a break from work, see the world and enjoy time with family. Adequate trip planning and preparation helps travelers accomplish trip goals safely and enjoyably. The trip planning includes finalizing a destination, making travel arrangements, finding an accommodation and deciding places to visit or activities to perform. One needs to dedicate ample of time for successful trip planning. One may have many questions during each phase of the planning. Travel sites like Orbitz, Expedia or Travelocity though may provide some information on the destinations but they may not have answers to each and every question of the traveler. Existing online forums like Experttravelanswers do discuss about travel related questions but the user has to manually read through each question and answer to resolve their doubt. This suggests that there is a need of travel domain specific question answering system which can give precise answers for the user's questions.

Question Answering system is an active research area in the field of Natural Language Processing. The goal of the system is to automatically answer questions posed by humans in natural language. The system designed can be either domain-specific or domain-independent. The basic architecture of Question Answering system comprises of following sub-modules: Question analysis, Document retrieval, Answer extraction, Answer evaluation and Answer display. The main focus of this project is the question analysis subtask which involves natural language understanding of the questions and its type identification. The question type identification task will accurately assign labels to the questions based on the expected answer type. For example, if the question is "Can I go for hiking in Chicago?" then it's question type will be ACTIVITIES. Question analysis task can be further divided into multiple sub-tasks i.e. preprocessing, POS tagging, parsing , N-gram computation, feature extraction, question classification and generic template generation for questions of each category.

The problem definition is interesting to us for two reasons. Currently as per our knowledge there is no existing travel domain-specific website or online forum that includes the above mentioned feature. Secondly traditional question answering systems analyze and label the questions by extracting keywords or by identifying patterns from them. Our proposed approach instead trains a classifier model using different features like N-grams and Part of Speech tag sequence to predict the label of the question. We assume that the correctness of the answer is highly correlated to the label assigned to the question.

2 Related Work

Till date, there have been a significant amount of research work on question classification, question taxonomies and question classifier. The annual Text Retrieval Conference (TREC) Challenge has also been able to attract lot of researchers to work on this field since the past decade. The approaches that have commonly been used for question classification can broadly be divided into three categories: rules based approach [1], machine learning approach [2] and hybrid approach [3]. Systems that use rule based approach work on a set of handcrafted rules to determine the type of answer expected. Machine learning approach based systems extract different kinds of features from the ques-

tions and use classification algorithms like Supervised Vector Machine, clustering, etc. to predict its type. Hybrid systems, on the other hand, use rules to generate features and then use classification algorithms on those features. There are also some research work that focus on specific domain, language and kind of questions. Banerjee et al. [4] have performed an extensive analysis on the various kinds of questions that are possible in Bengali language. They have extracted several lexical, syntactic and semantic features and have performed classification using Nave Bayes, Kernel Nave Bayes, Rule Induction and Decision tree Algorithms. Wang et al. [5] have developed a system for mobile service consulting area using ontology and question template based Question Answering System. Zukerman et al. [6], on the other hand, have performed classification of WH-questions and have evaluated the effect of question length and information need on the predictive performance of the system. Among the many other significant work that have been done on question classification, some of the most well-known ones are by Gharehchopogh et al. [7], Molino et al. [8] and Li et al. [9]. But to our knowledge, there is no such Question-Answering system for travel domain and this motivated us to develop a system that can classify questions related to travel, which can later be integrated into a travel domain specific Question-Answering system.

3 Corpus

We extensively researched several travel blogs, websites and forums in pursuit to find sufficient amount of query data. Out of all of them, we finalized two online forum sites, namely, Stack Exchange and Expert Travel Answers. In both forums, users post questions along with either user-defined labels or pre-defined labels from a drop-down list and get replies from other users or experts. The former had approximate 8000 and later had 400 travel related queries. We imported the dataset from Stack Exchange into a comma separated file by querying its database online. First column represented the title of the post and second column represented the list of tags assigned to it by the user. For example, "Must-see places in Goa?,<sightseeing ><India ><Goa >". We wrote a macro in Microsoft excel to download the data from Experttravelanswers online forum. There was separate excel sheet for queries pertaining to each category. For example,<outdoor adventure >category had 45 queries where as <couple >category had 118 queries. In spite of the dataset being annotated, we had two major issues. In many examples, the title of the post was not a question but a short descriptive text explaining user's problem. Secondly the tags were either too specific or personalized with respect to the user's question. For example, consider the above <couple >category. In this category, the queries were about inquiring about some destination or places to visit in a particular city. But as the questions were posted by married couples searching for vacation destinations, these questions were being categorized under<couple>category. In order to solve the first issue we had to re-frame the sentence if needed in form of a question in order to simulate the kind of questions a user might ask. This was important because we found a significant difference in the given post and the kind of question a user might ask to an automatic question answering system. Moreover for resolving the second issue, we extensively researched the various travel booking related websites mentioned earlier and analyzed the steps followed in booking a trip. We created a list of all possible words that could be assigned as tags to the questions. Figure 1 is the screen shot of the same.

Attractions/Culture/Events	Beach/water activities	Cruises	Getting there	Hotels/Lodging/Spas	Outdoor Adventure	Preparation	Food
Aquariums/Zoos	Beaches	Adventure/Expedition Cruises	Air travel	All-inclusive resorts	Biking	Packing/Luggage	Bar
Festivals/Events	Scuba/Snorkeling	Luxury/Premium Cruising and Cruise Lines	Bus travel	Beach resorts	Rafting	Passports/documents	Bakery
Museums/Historical	Fishing	Mainstream Cruises and Cruise Lines	Train travel	Boutique Hotels	Camping	Travel insurance	Restaurant
Performances	Boating	River Cruises and Cruise Lines	Car travel/ road trip	Hotel chains	Skiing/Snowboarding	visa	Café
Theme parks	island	Sailing Vessels/Yachts	Car rental	Spas	Hiking	multiple entry	cuisine
Bridge		Money-saving tips	Travel times/distances	Hostels	Eco-tourism	single entry	
Building		Shore Excursions	Help! I'm Stranded	Vacation Rentals/Swaps	National parks/monuments	visa free entry	
Book Shop				Money-saving tips	Bunjee_Jumping	working visa	
Cemetry				Accommodation	Safari	visa extensions	
Church				BedAndBreakfast	Sunbathing	souvenirs	
Cinema				Campground	Surfing	visa on arrival	
City wall				Luxury Hotel	carnival		
Club				AccomodationRating	excursion		
Comedy club				BudgetHotelDestination	trekking		
concert hall							
department store							
shopping mall							
Fort							
Ice Rink							
library							
memorial							

Figure 1: Sample list of keywords important in travel domain.

We then finalized our-defined 8 categories explained in Table1.

Therefore, we wrote a java program that helped us in manual annotation and storage of the modified dataset. The dataset comprised of many irrelevant questions. Moreover, the dataset being too big for manual annotation we wrote a java program to reduce the size of the dataset for manual annotation. We automatically identified city and country names using Named Entity Recognition and relabeled them as location. Secondly, we tried to select the data for annotation by matching the user-specified tags with the important keywords for travel domain. We selected those questions

Category Name and Description
Name:LOCATION Description: Questions related to location of a destination or questions inquiring about list of destinations Example: Where is Chicago located in the United Stated?
Name:ATTRACTION Description: Questions inquiring about places to visit in a particular destination Example: Is Eiffel tower worth visiting in Paris?
Name: ACTIVITIES Description: Questions inquiring about activities to do in a particular destination Example: How can I go for hiking in San Diego?
Name:TRAVEL PREP Description: Questions related to any kind of travel preparations like tours, important travel documents etc Example Do I need a transit visa for London from India to Chicago?
Name: ACCOMMODATION Description: Questions pertaining to finding an accommodation for the holiday stay Example: Can you suggest low budget hotels in New York?
Name: FOOD Description: Questions related to restaurants and bars Example: Are there any Japanese cuisine restaurants in Chicago?
Name: DEALS Description: Questions related to discounts, deals and packages Example: Are there any special offers for newly wedded couples while booking a hotel?
Name: INFORMATION Description: Questions inquiring about particular information other than the above mentioned categories Example: Can I get a rental car in London while traveling for a 3 day road-trip?

Table 1: Our defined categories

where at-least 2 of the tags matched the words from the topic list. The size of the reduced informative dataset was 2000. We then manually annotated each of them individually and calculated the Kappa score. It turned out to be 0.76. Figure 2 explains the distribution of the data as per the 8 categories. The size of the concerned dataset further reduced to 668, out of which we selected approximately 500 questions as the training dataset pertaining to LOCATION, ATTRACTION, ACTIVITIES, TRAVEL PREPARATION AND ACCOMMODATION categories. We ignored the remaining categories because the available samples for those categories were either very small or were of mixed quality. Below mentioned figure explains the distribution of the class labels over entire dataset.

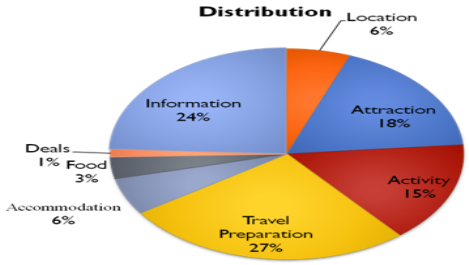

Figure2: Distribution of the class labels over entire dataset.

4 Approach

4.1 Topic Modeling

Assigning label to the question on basis of its expected answer type is in general a multi-class multi-label problem. For simplicity purpose, we assume that our problem definition is only a multi-class problem and have annotated the dataset accordingly. We proposed 8 generic category labels as explained earlier instead of too specific or personalized labels. In order to check the coherence of the labels with respect to the dataset, we performed topic modeling using MALLET and Stanford Topic Modeling tool. We experimentally performed unsupervised Latent Dirichlet Allocation (LDA) and supervised Labeled LDA topic modeling on the dataset. The number of topics ranged from 5 (Default) to 20 in LDA modeling and 5 in case of Labeled LDA. In case of LDA, though we obtained top topic keywords for the varied number of topics but it was very difficult for us to correlate our proposed topic labels to these topic keywords. As our dataset was now already annotated we also experimented with Labeled LDA topic modeling. There were significant number of relevant travel domain keywords representing our topic labels. This explained us that our proposed topics were consistent with the expected answer type of the corresponding annotated data. Figure 2 and Figure 3 provides the snapshot of the results of LDA and Labeled LDA models.

```
0   tour tours cost day exchange western fame shop disco
1   good passport things destination family long travell
2   cruise money foreign recommend traveling suggest loy
3   place trip luggage visiting companies insurance wort
4   travel bring type year top cruises week ships sea ca
5   visit island sites special pack time hiking car prog
6   vacation kind passport weather needed currency child
7   cruise port winter ship safe stop bands book visa lc
```

Figure 3: Topic keywords using LDA topic Model

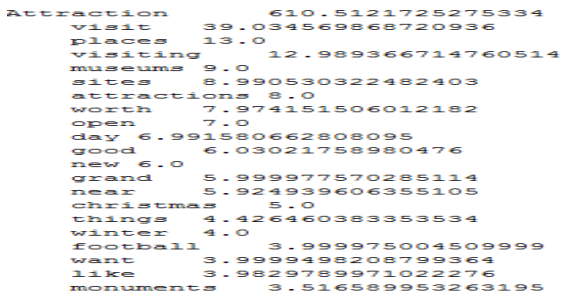

Figure 4: Topic keywords using Labeled LDA topic model

4.2 Feature Extraction

It is one of the crucial step before the classification task. We parsed each question with help of the Stanford parser and extracted two sets of N-gram features. One set comprised of complete list of unique unigrams, bigrams and Trigrams and another set comprised of selective N-grams based on their part of speech tag. In the second set of features, we have considered only those words that have POS tag from the following: either form of Verbs, Nouns, Adjective, Adverb, Wh-Determiner, Possessive Wh-pronoun and WRB Wh-adverb and Modal and have ignored the rest. Same is applicable for bigrams and trigrams. We calculate and store the term frequency counts (TF) instead of TF- IDF for each of the words in the two sets of N-gram features. Ideally, the relevant terms tend to have high term frequency and low document frequency. However, in our case the most relevant terms had lower values of TF-IDF. For example, verb like "visit" had lower TF-IDF in our dataset but it was very important keyword to identify Attraction related questions. On the other hand important keywords for identifying activity related questions like "hiking" and "skiing" ended up having a lower TF-IDF value instead of a higher one because of the small size of training set. As a result, we only considered the term frequency of each N-gram.

We also extracted the entire POS sequence of the sentence as a feature. In this case, we only considered tags related to verbs, nouns and Wh-Determiner, Possessive Wh-pronoun,Wh-adverb and Modal and ignored the rest of pos tags. Moreover, while extracting the sequence we considered multiple consecutive occurrences of the same POS tag as a single instance. This helped us to detect patterns in the POS sequence in each category of the data. Consider for example, "WhatWP isVBZ thereEX toTO seeVB inIN ChicagoNNP duringIN myPRP$ 10CD dayNN holidayNN tripNN inIN summerNN ?.". The sequence of POS tags as per our approach will be "WP VBZ VB NN".Thirdly, we formed different features combining N-grams with other N-grams and also by combining N-grams with the POS sequence of the sentence.

Above mentioned feature selection and extraction is thus quite different from traditional domain specific question analysis systems which focuses only on extracting keywords or templates for each kind of question. Our feature selection and extraction is quite similar to the feature extraction in question analysis using machine learning approach. Both the systems focus on N-grams and POS tags of the corresponding N-gram. The difference in our proposed approach lies in the selective N-grams and POS sequence generation. We tend to use generic OS sequence of the entire sentence as explained earlier instead of random or neighboring POS tags.

4.3 Classification Technique

Once the term counts were calculated for all kinds of features, we tried to convert the file comprising of feature vectors into the specific format required by tools like WEKA and MALLET. Thereafter, we performed various supervised classification algorithms like: Naive Bayes, NaiveBayes Multinomial, SVM, MultiClass classifier, MaxEnt and Logistic Regression. We also performed experiments using bagging, a machine learning ensemble meta-algorithm. We also experimented with AttributeSelective algorithm which trained the supervised learning base model based on selective features evaluated using Information gain.

5 Results and Analysis

5.1 Tools used for Classification task

Weka (Waikato Environment for Knowledge Analysis) is a popular free software suite of machine learning software written in Java, developed at the University of Waikato, New Zealand.It supports several standard data mining tasks, more specifically, data preprocessing, clustering, classification, regression, visualization, and feature selection. All of Weka's techniques are predicated on the assumption that the data is available as a single flat file or relation, where each data point is described by a fixed number of attributes (normally, numeric or nominal attributes. MALLET is an integrated collection of Java code useful for

statistical natural language processing, document classification, cluster analysis, information extraction, topic modeling and other machine learning applications to text.

5.2 Results

We considered 514 annotated questions for our experiment. We performed 10 fold cross validation for training and testing the supervised classification model using WEKA and MALLET tools. The liner classification algorithms were either used as it is or were optimized with selective attribute feature using information gain parameter. Moreover, bagging - an ensemble classifier that consists of several classifiers and outputs the class based on the outputs of these individual classifiers was also trained and tested. Accuracy, F-measure, Precision and Recall metrics were recorded for each of the algorithms along with the confusion matrix. The results were then analyzed. Table 2 indicated the accuracy obtained by each algorithm. The value in bracket indicate the accuracy achieved by optimizing the algorithm with selective attribute feature.

Features	NaiveBayes	LibSVM	Bagging	Multi-Class
Unigrams	80.50	72.94	82.55	74.49 [75.29]
Bigrams	65.69	49.90	69	[68.22]
Trigrams	-	-	55.36	-
Uni+Bigrams	86.66	67.25	**87.45**	-
Uni +POS sequence	71.76	70	83.14 [83.52]	-

Table 2: Percentage accuracy reported by various algorithms using respective features.

Class Name	Precision	Recall	F-measure
Accommodation	86.7	63.4	73.2
Activity	67.1	1	80.3
Attraction	98.2	84.4	90.8
Location	90	85.7	87.8
Travel Prep	1	87.5	87.9

Table 3: Detailed accuracy per class for Bagging ensemble classifier using Tree classification model with Unigram + Bigram features

a	b	c	d	e	classified as
26	12	0	3	0	a-accommodation
0	**110**	0	0	0	b-activity
1	18	**108**	1	0	c-attraction
2	4	0	**36**	0	d-location
1	20	2	0	**166**	e-travel prep

Table 4: Confusion matrix per class for Bagging ensemble classifier using Tree classification model with Unigram + Bigram features

5.3 Analysis

We carried out extensive experiments using different classifiers but we would be explaining analysis of just one of them due to limitation of the paper length. We would be submitting summaries of remaining results in text file with the code. Firstly, the accuracy and F-measure reported in the above results does agree with our prediction in spite of the problem definition being complex and the presence of few overlapping queries in two of the categories namely Attraction and Activity. According to both of us, we believe the results to be motivating considering the fact that it was our first attempt. As we could not find any existing travel domain specific question answering system, we were not able to do a comparative analysis of our approach. The highest percentage accuracy is achieved by using combined unigram and bigram features with bagging - ensemble classifier with a REPtree - a tree based learning algorithm as a base learner. We observed that bootstrap aggregating, also called bagging, gave the best accuracy irrespective of whichever feature is used. It is so, because bagging is designed to improve the stability and accuracy of machine learning algorithms used in statistical classification and regression. It reduces variance and helps to avoid overfitting. Moreover, REPTree is a fast decision tree learner which builds a decision/regression tree using information gain as the splitting criterion, and prunes it using reduced error pruning. It only sorts values for numeric attributes once. Missing values are dealt with using C4.5s method of using fractional instances. This explains the reason for obtaining high accuracy.

Secondly, we observed that SVM classifier though popular for text classification performed worst compared to other classifiers. We believe the possible reason for the failure is the imbal-

anced size of the different classes because SVM fails completely in situations with a high degree of imbalance. Here there is vast difference between the size of the data available for each class for training SVM model. Naive Bayes on the other hand gave better results. In order for Naive Bayes to achieve good results, the assumption of independence must be satisfied by the variables of the dataset and the degree of class overlapping must be small. In the available dataset, both the conditions were satisfied which explains the reason behind the obtained results. However, as there was certain degree of overlapping between the N-grams of different classes, bagging method helped us to improve the accuracy.

Thirdly, We predicted that there were chances of misclassification in Activity, Attraction and Travel Prep class due to presence of overlapping N-grams. It was then evident from the confusion matrix. The reason behind misclassification in Accommodation class is due to small size training dataset.

6 Conclusion

We have put our first step towards building a travel domain-specific question answering system. Our main focus for the project was Question Analysis. We have tried to assign labels to travel related questions based on the expected answer type. We used different variations of N-grams and POS sequence of the sentence as features to train various classification models. The combination feature of Unigrams and POS sequence of sentence is one of the novel approach suggested and tested by us. Apart from question analysis, we learnt the significance of annotating a dataset and its correlation to the development of a good question classfication model. We have tried to expolore different natural language techniques like parsing, named entity recognition,topic modeling and kappa statistics.

The limitation of our approach is that we have assumed our problem definition to be a multi-class classification problem instead of a multi-class multi-label problem. Secondly, we have also assumed that the input question will only be related to travel domain.

We would like to extend our approach for multi-label classification problem as well. Secondly, we would like to test combined feature of unigram, bigram and POS sequence. Though we have already extracted that feature but due to limitaiton of time we were not able to test it. Thirdly, we would also like to find out whether we can extract common template for questions in each category.

7 Appendix

We both started with extensive research on working of various travel related websites and online forums. Amruta imported data from StackExchange and Sabita wrote a macro in Excel to import data from ExpertTravelAnswers.com. Amruta wrote a java program to identify location entities from the data and also automatically select the questions assigned with the most relevant number of tags for manual annotation. Both of us then manually annotated the reduced selected dataset. Amruta wrote scripts to execute unsupervised LDA topic modeling and supervised labeled topic modeling in order to check the coherence between the manually assigned topics and automatically detected topics using MALLET and Stanford Topic Modeling tools. Meanwhile, Sabita calculated the Kappa score value and also wrote java programs to extract various N-gram features like Unigram, Bigram, Trigram and Unigram+Bigram. Further, Amruta wrote programs to extract Part Of Speech sequence of the entire sentence and also to convert the extracted features into the format required for tools like MALLET and WEKA for classification. Finally both of us together then trained, tested and analyzed various classification models using WEKA and MALLET.

8 References

[1] D.A. Hull, “Xerox TREC-8 question answering track report, in Voorhees and Harman, 1999.

[2] T. Nguyen, L. Nguyen, A. Shimazu, “Using semi-supervised learning for question Classification in Journal of Natural Language Processing, 2008.

[3] J. Silva, L. Coheur, A. Mendes and A. Wichert, “From symbolic to sub-symbolic information in question classification in Artificial Intelligence Review, 2011.

[4] S. Banerjee, S. Bandyopadhyay, “Bengali Question Classification: Towards Developing QA, in Proceedings of the 3rd Workshop on South and Southeast Asian Natural Language Processing, 2013, pp 2540.

[5] D.S. Wang ,“A Domain-Specific Question Answering System Based on Ontology and Question Templates, in Proceedings of the 11th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, 2011.

[6] I. Zukerman, E. Horvitz, “ Understanding WH-Questions: A Statistical Analysis”, in Proceedings of Association for Computational Linguistics, 2001.

[7] F. S. Gharehchopogh, Y. Lotfi,“Machine learning based Question Classification Methods in the Question Answering Systems”’, International Journal of Innovation and Applied Studies, 2013, pp 264-273.

[8] P. Molino, P. Basile, “QuestionCube: a framework for Question Answering”, in Proceedings of Central Europe Workshop, 2012, pp167-178.

[9] X. Li, D. Roth, “Learning Question Classifiers”, in Proceedings of 19th International Conference on Computational Linguistics, 2002.