MSiA 400

Q&A Project

Jessica Chan, Michael Cho, Annie Didier, Rohan Jha, Roshan Kumar

Table of Contents

Executive Summary

This project is based on answering frequently asked questions, which has various applications in real-world scenarios, ranging from replying to students' queries for a college application to answering a customer's question in an e-commerce setting. Much time is taken up by responding to the same types of question. Automating these processes allows the administration to devote time to efforts that generate business value.

The project was divided into four parts. Our data files come from MSiA administration. We were provided with two files containing questions (incoming emails from prospective students) and answers (outgoing emails from MSiA admin) in raw format. The first part was to match questions with their corresponding answers, and the second was to perform descriptive statistics on common attributes of questions that have many responses. Subsequently, we clustered questions into several categories to determine the FAQ. Finally, we identified answers for possible FAQ to generate template answers. We estimate that 66% of the questions can be addressed using a standardized response.

In the next section we describe our methodology for each of the different parts of the project, problems faced during the implementation and finally concluding with the result and some of the interesting findings.

Github: https://github.com/MSIA/Hayward.git

EDA and Data Cleansing

Indexing

We attached an index to each question and answer and stripped off escape characters (like \r, \n, \t) to prepare the data for preprocessing.

Information extraction

We applied a combination of regular expressions to trim and massage the data into a format that could be used for further processing. Relevant information from the emails like subject, body and name of the inquirer was extracted. The ‘Name’ parameter was especially interesting as there was no fixed behavioral pattern that could be exploited. The names had to be extracted from a plethora of sources like, subject of the email, greetings in the body of the email, email signature etc.

Merging

The questions and answers were then matched with each other based on a set of parameters. The key parameters were ‘Name’, ‘Subject’ & ‘Timestamp_count’. The challenge here was to handle scenarios in which there were multiple emails in a conversation chain, and we needed a reliable method to match each question to the specific answer response. We were able to determine email chains by extracting timestamps from emails.

Word Frequency (Data Visualization)

To gain further insight into the important topics discussed in the emails, we created histograms of the length of emails and FAQ categories, as well as word clouds of the most frequent words. These charts could not be run directly on the raw email data. We applied transformations like removing punctuations, removing commonly used words & word stemming so that we obtained meaningful visualizations.

Techniques Used

Sentence Splitter and Question Extraction

The content of each question email varies significantly. It can be as simple as a courtesy reply, or it can contain multiple sophisticated questions. To improve the accuracy of classification of questions, the emails are parsed into sentences rather than treated as a single amalgamated document. The sentence splitter marks up individual sentences using a set of heuristics for detecting the end of a sentence by a question mark, an exclamation mark, or a period.

Document Term Matrix

Before classification, we needed to prepare a clean corpus for text analysis. The cleaning process included (not in any particular order): converting the text to lowercase, stemming, and removing punctuation, numbers, and extra whitespaces between words. Next, we created a Document Term Matrix, which is a matrix with documents (emails) as rows and words as columns and the frequency of words appeared in the documents for each entry. We removed sparse terms and created a word cloud for visualization. We attempted to use the same DTM for cluster analysis, but this resulted in clusters whose words were not meaningfully combined. To overcome this, we created another Document Term Matrix, with values signifying relative importance/relevance to documents using Tf-Idf weights instead of plain frequency to better assign terms to documents.

Classification and k-means clustering

In order to find the frequently asked questions, we needed to find clusters of similar questions. We used the k-means clustering method to measure closeness of questions/documents using the Tf-Idf weighted DTM. The choice of k was initially based on where the graph ‘bends’ in a plot of the sum of squared error as a function of k.

However, this resulted in a high number of clusters that did not contain meaningful content. Another problem we encountered was, even if we use a large number for k, there were still one to two clusters that contained ⅔ of the documents, and within these big clusters, there were obviously disparate categories. We decided that number of clusters was too high and needed to better address the needs of the client. Since there are 15 FAQ on the MSiA website, but Vicky and Sarah still got emails, we decided to expand this to 20.

Keywords and Categories

Within each cluster, we calculated the relative closeness of each term to their respective centers and provided a score to each term document pair. We then took a difference between the score for a particular cluster to the average scores of the remaining clusters for each term and extracted the highest ten keyword scores. We analyzed these top keywords by looking at their frequency in the documents, semantic context, and word associations. Thereby, we were able to come up with categories (FAQs) that are present in the emails. We also applied the same type of categorization to the answer emails. After splitting the answer emails into sentences, we sorted which sentences (response) by frequency for each category and then devised the most appropriate template answer.

Conclusion

K-means clustering produced a grouping of top ten keywords for each cluster, which did not necessarily fit semantically in one sentence. We had to manually create categories and match them to template answers by finding the frequent response sentences.

Our methodology required a fair amount of manual labor and heuristic approaches. For instance, the value for k that we chose based on statistical methods did not result in meaningful output, so we had to pare down the value of k based on the number of

topics we thought would be meaningful to the client. Additionally, when we determined the most frequent answer sentences, responses such as “I hope this was helpful”, were at the top and needed to be removed.

The clustering and classification methods enabled us to digest a massive data set that would prove difficult otherwise to analyze and derive meaning from . However, it would be useful to to further automate the process of generating template responses so that human intervention is not required or at the very least more limited.

References

Hands-On Data Science with R Text MiningGraham.Williams@togaware.com10th January 2016 Visit http://HandsOnDataScience.com/ for more Chapters.