SANTA CLARA UNIVERSITY

COEN 281 PATTERN RECOGNITION AND DATA MINING

TERM PROJECT PROPOSAL

Finding Topics That Have Limited Supports on Stack Overflow

Author
Ting-yu YEH
Nicholas FONG
Christian AYSCUE
Bing TANG

Supervisor
Dr. Ming-Hwa WANG

November 7, 2017

Contents

List of Figures

List of Tables

Finding Topics That Have Limited Supports on Stack Overflow

Ting-yu Yeh, Nicholas Fong, Christian Ayscue, and Bing Tang

Abstract— **The abstract will be further refined after we have the prototype. A text mining approach can be used on the posts on stackoverflow.com to find popular topics that do not have a lot of support for a given time period. By comparing these findings to job postings at later dates, we find patterns that predict what the later job postings will be based on earlier Stack Overflow questions.**

I. Introduction

Computer technology changes at a rapid rate. In order to keep up with the change, engineers must constantly be learning. However, developer time is a high-demand commodity, and so learning should be made efficient. Through Stack Overflow question posts and job board posts, patterns can be deduced to predict which skills will be most needed in the future. With this information, computer and software engineers can stay ahead of the curve with what they spend their time learning.

The objective of this paper is to predict the number of future job openings in a field by analyzing the current support for questions in that field on stack overflow. This information is critical to help developers know the skills that they will need for the future job market. The findings of our research should be useful to young developers who hope to find a job. With our findings and methodology, they can know which fields of computing they should become proficient in to land a job and prepare for the future. In this project, we are analyzing a large dataset from Stack Overflow Q&A entries with data mining techniques to uncover common themes across a massive number of text documents.

After research and discussion, we decide to use LDA (Latent Dirichlet Allocation) as the main categorization method to find the most popular topics because comparing to TFIDF (Term FrequencyInverse Document Frequency), which only finds individual terms among the posts, LDA is more versatile to identify topics by finding groups of words that frequently appear together. This approach allows us to categorize posts by topics where multiple topic-related words appear in the text together. This is much better for categorizing post topics than looking for the presence of individual terms is. The scope of our investigation is around topics that software developers encounter. Such topics are discussed primarily on stackoverflow.com, which is the data source we will use. We hope to categorize common topics by their level of answer support and relate this to the current job market.

T. Yeh, N. Fong, C. Ayscue, and B. Tang are with the Department of Computer Engineering, Santa Clara University, Santa Clara, CA, 95053 USA

II. Theoretical Bases and Literature Review

The problem we are trying to tackle is the identification of topics that have limited support on Stack Overflow.

A. Theoretical background of the problem

Topic identification in natural language processing is commonly done with Latent Dirichlet Allocation (LDA). LDA is an unsupervised clustering algorithm for natural language processing. It identifies primary topics based on the assumption that each document is composed of a mixture of topics that generate keywords. LDA matches frequently co-occurring keywords by iteratively improving its guesses. Once the solution converges, humans can manually label each topic based on its keywords.

B. Related research to solve the problem

A large body of researchers have investigated the nature of Stack Overflow. For example, Cristoffer Rosen and Emad Shihab have researched what mobile developers were asking on Stack Overflow [1]. Rosen and Shihab used Latent Dirichlet allocation (LDA) to cluster questions and then examined various statistics related to those questions such as response time, average number of answers, or the ratio between question views and the percentage of questions with accepted answers. Other papers have also implemented LDA, such as Manipal University's examination of question and answer quality [2]. Other work exploring the non-functional requirements for developers on Stack Overflow have also used LDA to cluster questions and then applied their analysis on each topic. As we can see, there is a large body of research using LDA to study Stack Overflow questions. Research is done not only on questions on Stack Overflow, but also the answers. Calefato et al. have done extensive research on understanding what makes a good answer [3]. They evaluated a good answer as the answer accepted by the asker as acceptable. They used Alternating Decision Trees (ADT), a binary classifier, to classify responses as either accepted or not. Using various features of the data, they were able to achieve 90% accuracy in classifying answers. They also found that counting number of votes on the answers alone is a better (with 93% accuracy) yet simpler metric to predict the accepted answer.

C. Advantage/disadvantage of those research

One advantage of the research using LDA is that it naturally finds topics that people are asking about on Stack Overflow. This reduces human bias in clustering questions into topics. LDA is also generally better than TF-IDF in terms of parsing the data. One advantage of using just the question titles is that it reduces the noise in each topic. One disadvantage though is that some questions can be misclassified since it doesn't include information contained in the body of the question. It was also helpful that the authors examined various features of the data to compare topics, such as the ratio of views to the percentage of accepted responses. These statistics allow us to see trends and draw insights about each different topic. However, the statistics analyzed only go so far, and there is information to be mined that hasn't been explored yet. An advantage of the ADT that they used to analyze the responses is that they were able to draw insights from the answers alone. However, the disadvantage is that for Stack Exchange sites, there is no use of such method, user votes for answers are proved a more efficient and more accurate way to evaluate the quality of answers.

D. Solution to solve this problem

Our solution to identifying topics that have limited support on Stack Overflow is a multi-step process. First, we will identify topics using LDA. Then, we will examine statistics related to each topic to try to identify which topics have the least support. We will identify such topics by evaluating "supply" vs. "demand". We wish to examine various metrics to analyze the "supply" and "demand" from the questions. We will see if characterizing demand as the number of question views, the number of question upvotes, or the number of questions allows us to draw insights not seen previously. Similarly, we will examine characterizing supply as the number of answers, the number of answer upvotes, or the number of unique responders allows us to see insights in the data.

Our solution differs from the papers examined in the ways that we evaluate whether topics have limited support. While some of the papers did broad analysis on a wide range of topics, or have done detailed analysis on various other features of the data, only we will provide such a detailed examination of the supply and demand in the questions and answers. An especially unique feature we want to examine is the number of unique responders, since few responders indicate that there are only a few experts in the field.

Our solution is better because we examine statistics not previously analyzed, statistics that are more relevant in determining which topics have limited support on Stack Overflow. Our utilization of data previously unexplored will allow us to draw better insights into Stack Overflow and the developer community at large.

III. Hypothesis

With the assumption that most working IT professionals seek out solutions and asked questions on Stack Overflow, we believe that the Q&A entries on Stack Overflow could represent a demand of skills in the IT industry. By analyzing the qualities and quantities of the entries on Stack Overflow, we believe that we could apply data mining techniques to extract information regarding future job trends. Therefore, we propose that the level of support for a field on Stack Overflow could predict job demand for that specific field in the future.

IV. Methodology

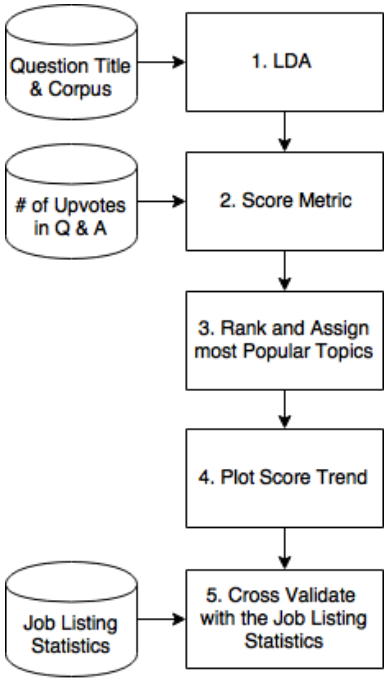

Fig. 1. General flowchart of the proposed algorithm

A. Data collection

To analyze the score trend of the selected topics. We download the Q&A entries from Stack Overflow and US technology job listing dataset from Kaggle.com. The file was separated by year, starting from 2008 to 2014. The title and corpus of the question entries were used for topic categorization with LDA. The upvotes of questions and upvotes of answers were extracted to calculate the score.

B. Algorithm design

The flowchart of the algorithm is shown in figure 1. The first step of the approach is to apply LDA algorithm to the titles and corpus of question entries. The granularity constant K will be tuned to lower the variance of

the topic numbers in each category. With the generated unnamed categories, the second step incorporated the votes of the questions and answers to calculate scores for each category. In the third step, the categories with the top scores of all times from 2008 to 2014 were named according to the LDA most frequent keywords. Only the top scored topics were plotted in the trend comparison. The score trend for the top ranked topics were plotted from 2008 to 2014. In our hypotheses, the score of the topics should match to the future popularities of the corresponding jobs.

We will mainly use python in this course project. Multiple open source packages would be used, including but not limited to pandas, NLTK, stop_words, and genism. We will use the online tutorial "Latent Dirichlet Allocation (LDA) with Python" [4] as a reference to apply the LDA algorithm.

C. Scoring metric

The scoring of the topics was calculated based on the supply and demand of the skills. High score indicates that the topic is popular with limited skill support. The score metric formula was designed as:

$$Score = \frac{Demand - Supply}{n} \tag{1}$$

Demand is the number of upvotes in questions, $Supply$ is the number of upvotes in Answers, and n is the number of entries, which normalizes the score for that specific topic.

The voting system in Stack Overflow subtracted the downvotes from the upvotes. The cancellation effect automatically takes the downvotes into account. The formula is based on the assumption that the number of question upvotes could fully represent the demand while the number of answer upvotes represent the supply of talent in that specific topic.

D. Testing against hypothesis

To test against the hypothesis, we will check if the rank of the topic will predict the future popularity of job fields by comparing the shape of the score curves with the job listing trends. Currently we have not decided on the source of the job listing statistics because the job titles have to fit in the categories summarized by the keywords generated from the LDA analysis. We will start searching the data when we obtain the results of the most popular topics from the LDA.

REFERENCES

[1] Christoffer Rosen and Emad Shihab, *What are mobile developers asking about? A large scale study using stack overflow*. Empirical Software Engineering, 2016

[2] R.K. Ranjitha and Sanjay Singh, *Is Stack Overflow Overflowing With Questions and Tags*. ACM Digital Library, 2015

[3] Fabio Calefato, Filippo Lanubile, and Nicole Novielli, *Moving to Stack Overflow: Best-Answer Prediction in Legacy Developer Forums*. ACM Digital Library, 2016

[4] Jordan Barber (2017, Nov. 6), *Latent Dirichlet Allocation (LDA) with Python*. Retrived from https://rstudio-pubs-static.s3.amazonaws.com/79360_850b2a69980c4488b1db95987a24867a.html