

Jarvis - A Smart Bot for Piazza

Pranav Ramarao, Isaac Bowen, Yu Ke
University of Michigan
Ann Arbor
{pranavr,irbowen,yuke}@umich.edu

ABSTRACT

In this work, we explore some of the challenges faced in Piazza specially for large sized classes. We present a smart bot - Jarvis that can tackle the problems by using state-of-the-art IR techniques. We show the Jarvis can greatly increase the efficiency of the instructors and also serve as a useful aid for students.

KEYWORDS

IR, piazza, search, similarity, FAQ generation

ACM Reference format:

Pranav Ramarao, Isaac Bowen, Yu Ke. 1997. Jarvis - A Smart Bot for Piazza. In *Proceedings of ACM Woodstock conference, El Paso, Texas USA, July 1997* (WOODSTOCK'97), 2 pages.
DOI: 10.475/123_4

1 PROBLEM

Piazza is a great tool for students and instructors to interact with teach other. It allows students to post questions on a class wide forum, where other students and instructors can answer the questions. For small classes, where the question volume is very low, works well. Instructors can respond quickly, and scale isn't an issue. However, as the departments grow and class sizes increase, it becomes difficult to answer all the questions quickly. For our project, we wanted to focus on a few specific issues related to Piazza at scale. The authors our instructors for undergraduate courses at UofM - these course have roughly 400 and 600 students, and only and handful of staff members to manage Piazza. We are interested in solving these problems to assists in our courses.

1.1 Duplicate Questions

As one of the side effects of the number of students posting questions, it becomes difficult for other students in the class to know if their question as already been asked. Because of the question volume, they may not have time to read every question that is asked, and as a result, repost a question that is very similar to one that has already been asked. For instructors, this means that they have to answer the same question over and over, or link students to the previous question. For students, it could provide confusion, as many very similar questions have very similar answers, but students are quite sure if its the same question (and therefore has the same answer).

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

WOODSTOCK'97, El Paso, Texas USA

© 2016 Copyright held by the owner/author(s). 123-4567-24-567/08/06...\$15.00

DOI: 10.475/123_4

1.2 Top Questions

A related issue is that students aren't getting the most they could out of Piazza because of the number of questions. Students can't dedicate hours every day to read through all questions posted every day. Also, many of the question are very student-specific (they won't related to other students, so the whole class doesn't benefit from reading them). However, there are some questions that are very common, or exceptionally helpful, and it would be beneficial to aggregate these high quality questions.

1.3 Better Search Features

Piazza's search features leave something to be desired. The largest problem seems to be that the results are very heavily skewed towards the most recent results, when that may not actually be what the student is looking for. We wanted to provide a method for students to more accurately search Piazza for their questions.

2 INTRODUCTION

Piazza is a free online gathering place where students can ask and answer questions 24/7 under the guidance of their instructors. Collaboration occurs in real time, which encourages participation and fuels discussion. Leading campuses across North America have a significant Piazza presence. Larger courses typically have over 500 students getting involved in the feed. Some of the painpoints however are :

- (1) students tend to ask duplicate questions without searching for them first.
- (2) students are overwhelmed by the huge number of questions on piazza specially in larger courses.
- (3) the search facilities that piazza provides are average and rely on recency based features and fail to consider relevance based ones.

2.1 Piazza

2.2 Chat Bots

3 METHOD

We designed and implemented Jarvis, a bot written in python that can accomplish these goals.

3.1 System Overview

Jarvis works by indexing all of the post on a given course on Piazza. On startup, it will bulk download all of the questions posted previously and constructed an index from there. After that, it sits in a loop, waiting for new questions to be posted.

3.2 Tools and packages used

We took advantage of several python packages and libraries to make the development of Jarvis faster and the results more correct. We used python whoosh for indexing the documents. This allowed us to index the documents, but first we actually had to collect them in one place. To do this, we needed to download all of the questions off of Piazza. Piazza does not have an official supported API, but there is an unofficial API that has been constructed through some reverse engineering. While this did not work as perfectly as a fully supported first party API would have, the results were surprisingly good and required only a small amount of work on our part to work with.

3.3 Data Extraction

Piazza does not provide an elegant or easy way to problematically interact with its posts. To get the questions and answers on the site, we had to use an unofficial API. This API has been reversed engineered by doing some things to fix it. We started from this API, and then built our own, higher level wrapper that would read in posts and then store them as objects in memory that were easier to work with. To speed up processing, testing, and to prevent Piazza from potentially rate limiting us, we also developed a feature to bulk download and store course data as a .csv file, which we could then treat as our "Piazza" input data for indexing and pre-processing purposes.

3.4 Indexing and Feature Extraction

3.5 Data Preprocessing

3.6 Duplicate Detection

3.7 Top Questions

3.8 Improved Search Feature

REFERENCES