A Programming Project Advisor

Maneet G, Mansi M, Neetha R, Aditi M, Vijaysri L, Harsh K
Georgia Institute of Technology, Atlanta, Georgia, USA

1. Motivation

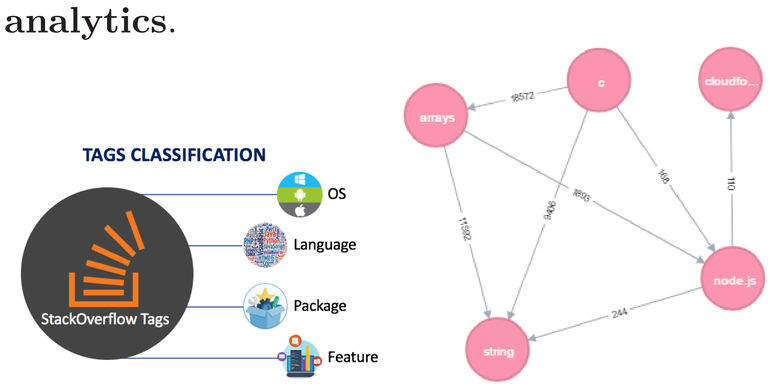
In today's world of constantly evolving technology, choosing the right programming tools to build a project can be daunting. Users need to Google, search StackOverFlow (SO), GitHub etc. separately to implement programming projects. With Pogo, users have a one-stop application which speeds up their learning.

2. Our Approach

- On a user's programming task query, Pogo offers insights into relevant technologies.
- Each technology is accompanied with its popularity score, degree of assistance and its dependencies.
- These insights help users make informed decisions on technologies best suited for them.
- Pogo thereby reduces time and effort spent in the planning and learning phase of a project.

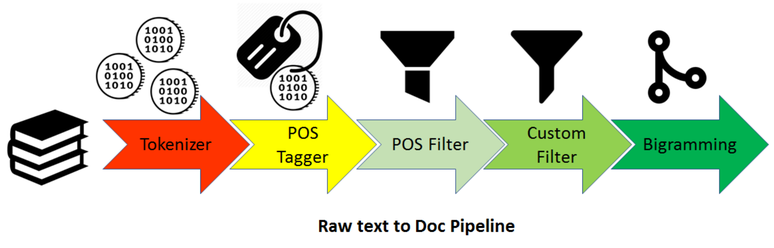
3. Data Preprocessing

Pogo is built by fetching SO, Libraries.io and GitHub data from Google BigQuery and integrating them using **topic modeling** and **graph analytics**.



Frequently occurring tags are classified both programatically and manually into 5 categories which helps in populating the user input/query form via exploiting a graph database in the back-end.

The questions data from Stack Overflow is fed to a preprocessing script which filters out irrelevant words via NLP processing (using *Spacy*) & custom filtering and performs bigramming on the filtered content. The 'docs' (list of key words) so generated are our main input to the topic model (LDA).

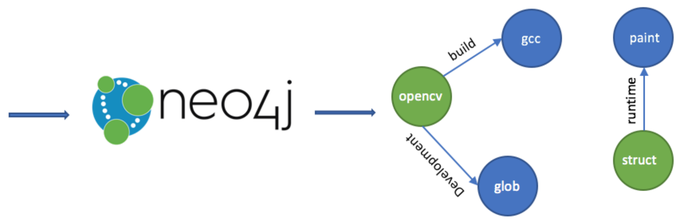


Example 1: "Any idea of deserialize mixed json string in gson?"
idea, deserialize, mix, json_string, gson
Example 2: "How to project Lat/Lon coordinates on a panorama image?"
lat_lon, panorama, image

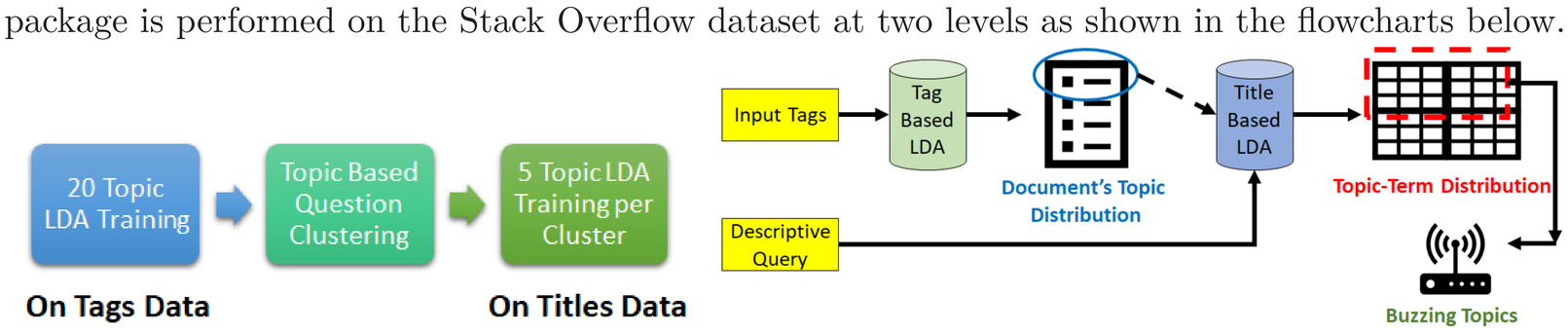
4. Training Strategies

Graph Analytics
Using the Libraries.io's *dependencies* dataset, we created a graph database in Neo4j. The graph comprises of project names as nodes and the project dependency names as edges. On a package name entered by the user, we query this database and display a graph depicting some of the package's dependencies (runtime, development, build, import, etc.).

Project name	Dependency name	Dependency kind
struct	paint	runtime
opencv	gcc	build
opencv	glob	Development



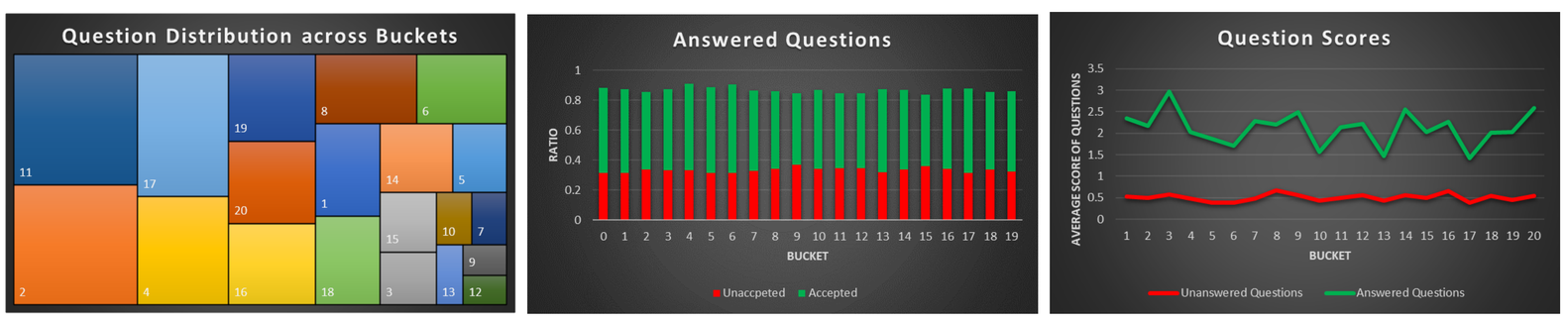
Topic Modeling
Topic Modeling using the *Latent Dirichlet Association* (LDA) algorithm via the *gensim* Python package is performed on the Stack Overflow dataset at two levels as shown in the flowcharts below.



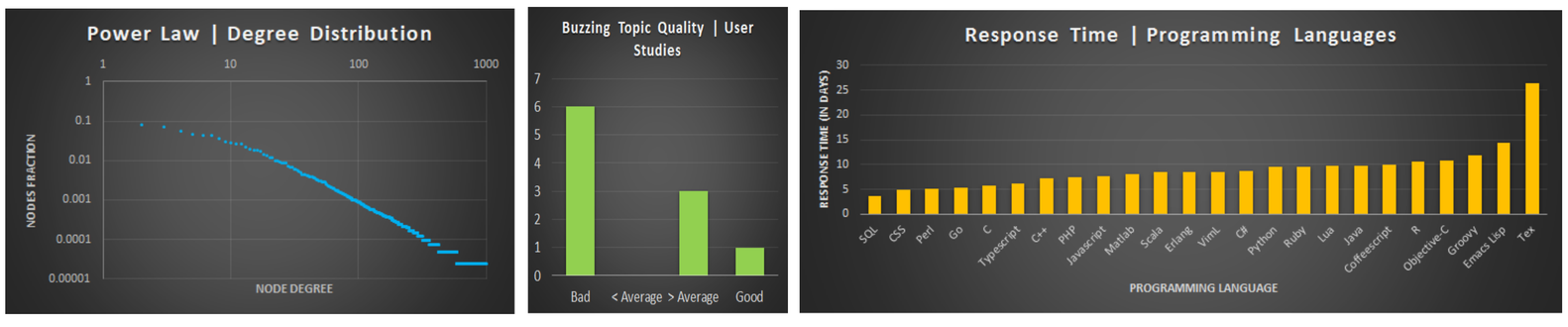
5. Features of Pogo

- **Assistance Score**: Represents the degree of assistance a user can expect from the SO portal. Is a function of the proportion of questions with accepted answers, quality of the concerned questions and answers, response time, etc.
- **Repository Reliability Score**- Based on the quality, popularity and number of issues a repository has, it is assigned a score indicating its reliability.
- **Buzzing topics**: A list of the most common topics that users tend to have questions in, while working on a given technology.
- **Smart and Friendly GUI**: Offers a quick and convenient path to our users to input their query (via dropdowns) and help them in improving their queries via suggesting frequently co-occurring input tags.

6.Experiments & Evaluation



Based on the LDA model output, we performed an exploratory analysis to understand the characteristics of our question buckets, for example, Bucket Size, Response Times, Answered/Unanswered questions and their quality. Bucket Size is helpful in gaining a better understanding of the Questions' Distribution across the entire dataset. A comparison of scores corresponding to answered and unanswered questions highlights the importance of their quality in influencing the likelihood of getting an answer on Stack Overflow.



A brief user study helped test the quality of buzzing topics generated by our suite of LDA models. And the tag co-occurrence graph network was successfully verified for degree distrib. power law.