# Job Market Dynamics: A Comparative Analysis of Trends and Skills Demand Across Online Platforms

Vikram Sawaram Choudhary
vchoudhary@binghamton.edu
State University of New York at Binghamton
Binghamton, NY, USA

Saurabh Patidar
spatidar@binghamton.edu
State University of New York at Binghamton
Binghamton, NY, USA

Rushikesh Eknath Bhadane
rbhadane@binghamton.edu
State University of New York at Binghamton
Binghamton, NY, USA

## Abstract

This report presents the implementation of a data collection system designed to analyze job market dynamics based on discussions from Reddit and 4chan. The system continuously collects posts and comments related to employment, skills, and job opportunities. The collected data is used for sentiment analysis, skill demand analysis, and trend identification. Key challenges addressed include API limitations, thread lifespan, and managing concurrency. The report also includes a preliminary exploration of the data, updated data projections, and a plot showing data collection trends over time. Insights from this analysis are valuable for job seekers, employers, and policymakers to understand workforce dynamics.

**ACM Reference Format:**
Vikram Sawaram Choudhary, Saurabh Patidar, and Rushikesh Eknath Bhadane. . Job Market Dynamics: A Comparative Analysis of Trends and Skills Demand Across Online Platforms. In . ACM, New York, NY, USA, 3 pages. https://doi.org/101

## 1 Introduction

This project aims to analyze job market dynamics by collecting discussions related to job opportunities, skills, and employment from Reddit and 4chan. The focus is on leveraging sentiment analysis, skill demand analysis, temporal trends, and network analysis to uncover key patterns in job-related discussions. By examining community-driven content, the project provides insights into the evolving job market, including sentiment towards various job sectors, recruitment challenges, and in-demand skills. The collected data is used to identify key influencers and patterns of job discussions, which can help policymakers, job seekers, and employers better understand the dynamics of the workforce.

## 2 Implementation Overview

The data collection process was implemented using Python scripts that interact with APIs from Reddit and 4chan. For Reddit, the solution used the Reddit API to collect posts and comments from multiple subreddits, while for 4chan, a RESTful API was utilized to gather data from relevant boards. The project included the following key components:

### 2.1 Reddit Crawler

The Reddit crawler interacts with the Reddit API to collect posts and comments from multiple subreddits. The flow begins by fetching subreddit data, extracting the necessary fields from posts, and storing them in a MongoDB database. Job scheduling is handled by Faktory, which ensures continuous data collection. Comments are fetched for each post using separate jobs, and all collected data is stored for further analysis. The system runs continuously, managing requests efficiently to avoid hitting Reddit's rate limits.

### 2.2 4chan Crawler

The 4chan crawler uses a RESTful API to retrieve archived data from specific boards, focusing on job-related discussions. The flow involves building API requests to either fetch the catalog or individual threads, identifying active and archived threads, and storing the extracted metadata in MongoDB. Faktory manages the crawling tasks, ensuring that each thread is processed in parallel and archived as needed. The system continuously monitors the /g/ board to capture ongoing discussions, ensuring that data is collected in real time.

The data collection system has been designed to continuously extract and store data from Reddit and 4chan, allowing for ongoing monitoring of job-related discussions.

## 3 System Description & Design

The data collection system comprises several interconnected components, illustrated in the system architecture diagram (see Figure 1).

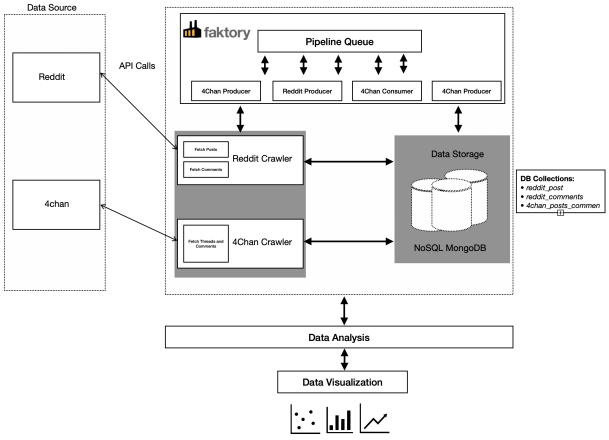


**Figure 1: Project Pipeline**

.
*Social Media Project Proposal, Sept 20, 2024, Binghamton, NY, USA*

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/101

### 3.1 Reddit Crawler System Operation

- The main script registers consumers for fetchPost and fetchComments jobs and creates an initial fetchPost job.
- The fetchposts handler creates a new fetchPost job with a 5-minute delay. It fetches a list of subreddits from a comma-separated text file, retrieves new posts for each subreddit, stores the data in the database, and creates a fetchComments job for each post.
- The fetchcomments handler fetches all comments for the given posts, processes them, and stores them in the database.
- New comments on posts are continuously monitored for the 30 latest posts in each subreddit, with the option to increase this number.

### 3.2 4Chan Crawler System Operation

- Initialization and Consumer Setup: The main controller initializes configurations, sets up logging, registers job consumers for fetchCatalog and fetchThread, and schedules the initial catalog fetching job.
- Periodic Catalog Fetching: The fetchCatalog handler periodically retrieves the latest thread catalogs from specified 4chan boards, identifying and storing new threads in the database.
- Thread Data Retrieval: For each new thread, a fetchThread job is created to fetch all posts within that thread, processing and storing the post data in MongoDB.
- Continuous Monitoring: The system continuously monitors the most recent threads on each board, scheduling regular updates to capture new posts and maintain up-to-date data.
- Robust Logging and Scalability: Implements comprehensive logging for tracking operations and errors, handles exceptions gracefully, and leverages a job queue system (e.g., Faktory) to ensure scalability and efficient processing.

The overall system architecture consists of:

- **Data Collection:** Python scripts that interact with Reddit and 4chan APIs to retrieve post, comment, and thread data.
- **Job Management:** Faktory, which manages asynchronous crawling jobs. The data collection scripts utilize Faktory to distribute tasks for crawling threads and collecting data at regular intervals.
- **Data Storage:** MongoDB is used to store the collected data. Each post, comment, and thread is stored as a document in the appropriate collection.
- **Docker Compose:** A docker-compose.yml file is used to set up and manage services for MongoDB, Mongo Express, and Faktory, ensuring seamless integration and easy deployment.

## 4 Data Collection Process

Data collection is done in real time, capturing discussions as they happen. The Reddit crawler involves fetching posts and comments from multiple subreddits such as r/jobs, r/recruitinghell, r/cscareerquestions, and r/technology. For 4chan, the crawler builds API requests to fetch either the catalog or specific thread details. The collection process involves:

- Retrieving the catalog for the /g/ board.
- Identifying new threads and distinguishing between active and archived threads.
- Crawling threads for posts, extracting relevant metadata (author, content, timestamp, etc.), and storing the data in MongoDB.

## 5 Challenges & Solutions

Several challenges arose during implementation, including:

- **API Limitations:** 4chan's RESTful API is limited compared to more robust APIs like Reddit's. As a workaround, the implementation relied on optimized HTTP requests and retry mechanisms to ensure continuous data collection.
- **Thread Lifespan:** 4chan threads have a short lifespan. To handle this, the system continuously monitors threads and archives them as soon as they become inactive.
- **Concurrency Management:** Managing concurrency was crucial to ensure the system's efficiency. Faktory was used for this purpose, allowing parallel crawling tasks to be handled smoothly.
- **Reddit Rate Limits:** Reddit imposes rate limits on API requests. To address this, the system schedules requests strategically to avoid hitting these limits while ensuring timely data collection.

## 6 Preliminary Exploration of Data

An initial exploration of the collected data has revealed the following:

- The number of threads and posts collected from the /g/ board varied significantly depending on ongoing discussions and external events.
- The Reddit data showed frequent discussions around career opportunities, technology trends, and issues faced in recruiting processes.
- Topics such as skill requirements and programming-related advice were prevalent across both platforms.

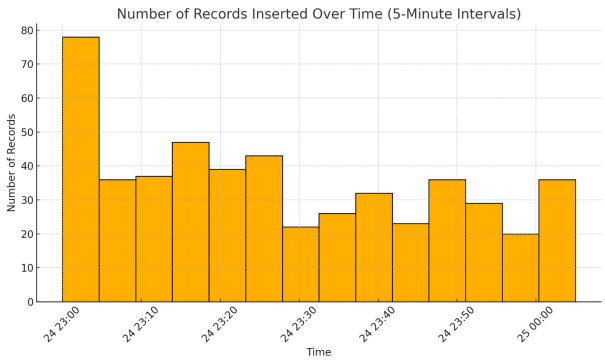


**Figure 2: Reddit Data Volume**

Figure 2 and 3 illustrates a preliminary plot showing the volume of data collected over time. Spikes in activity were observed during major technological announcements or layoffs.

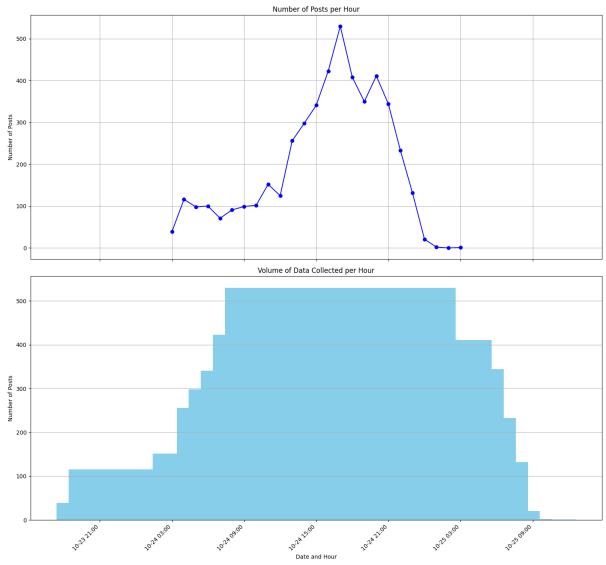


**Figure 3: 4Chan Data Volume**

## 7 Data Volume & Updated Projections

For Reddit, approximately 1.4 MB of data is being collected per hour on average.

- Weekly Estimate: 1.4 MB/hour × 24 × 7 = 235.2 MB/week

For 4chan, approximately 75 KB of data is being collected per hour on average.

- Weekly Estimate: 75 KB/hour × 24 × 7 = 12.6 MB/week

## 8 Reference Links

### 8.1 Reddit References

- **Reddit API Documentation:** https://www.reddit.com/dev/api/
- **Faktory Documentation:** https://contribsys.com/faktory/

Reddit APIs:

- **Access Token:** /api/v1/access_token
- **Fetch Posts:** /oauth.reddit.com/r/subreddit/new
- **Fetch Comments:** /oauth.reddit.com/comments/post_id

### 8.2 4Chan References

- **4chan Documentation by SSRC:** https://copeid.ssrc.msstate.edu/wp-content/uploads/2022/06/FINAL-4chan-Documentation.pdf
- **4chan API Documentation:** https://github.com/4chan/4chan-API/blob/master/pages/Threads.md

4Chan APIs:

- **Retrieve Catalog:** https://a.4cdn.org/board/catalog.json
- **Retrieve Thread Posts:** https://a.4cdn.org/board/thread/thread_id.json
- **Access Token:** /api/v1/access_token
- **Fetch Posts:** /oauth.reddit.com/r/subreddit/new
- **Fetch Comments:** /oauth.reddit.com/comments/post_id

## 9 Conclusion & Future Work

The implemented Reddit and 4chan crawlers successfully collect and store job-related discussions in real time, providing a valuable data source for analyzing the job market. Moving forward, the next steps include:

- Expanding data collection to include more boards and subreddits to broaden the scope of analysis.
- Implementing sentiment analysis on the collected data to understand community sentiment towards job market trends.
- Developing more sophisticated visualizations using React to display trends and insights interactively.

This project lays a foundation for understanding how online communities perceive the job market, providing actionable insights for different stakeholders.