



AI-based Model for Closed Job Detection

Mentor: Yusi Zhang, SongTao Guo



Zixuan Li

Software Engineering Intern 21'



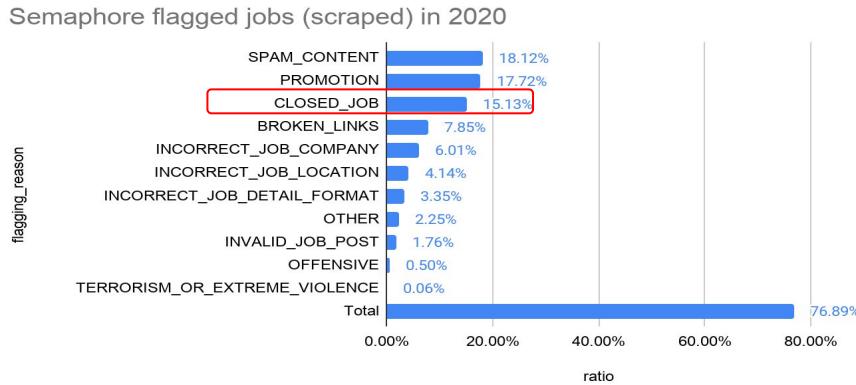


Agenda

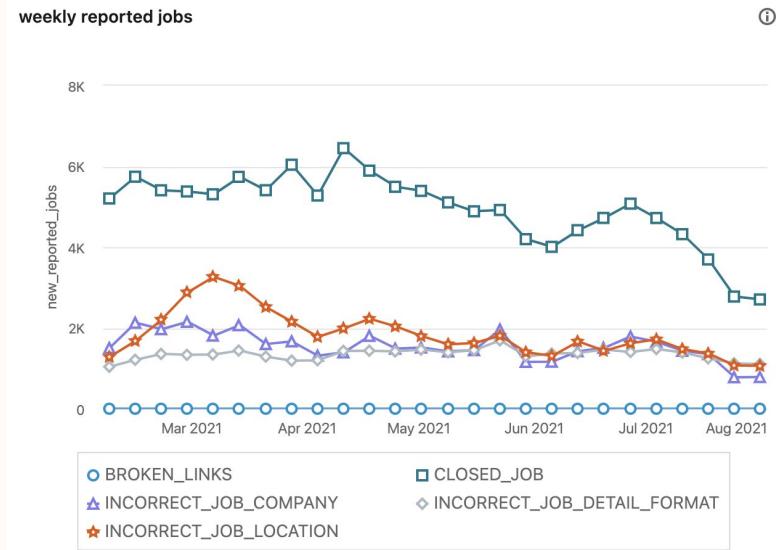
- 1 Problem Statement
- 2 Data Preparation & Feature Extraction
- 3 Modeling & Evaluation
- 4 Challenges & Takeaways

15%+

Of user complaints in 2020 is contributed by closed jobs



CLOSED_JOB is one of the top job quality issues of the ingested jobs reported by our members.



$$P(s(j) = \text{closed} \mid \langle j, p_t, c_t \rangle) = ?$$

$$f: \mathbb{R}^n \rightarrow \{\text{open}(0), \text{closed}(1)\}$$

Binary Classification Problem

Footnotes

1. j as job from all ingested jobs with external apply links
2. $s(j)$ as binary status of job j
3. p_t as web content of apply page p given time t
4. c_t as context given time t

Closed/Open Jobs

examples

joblift.fr

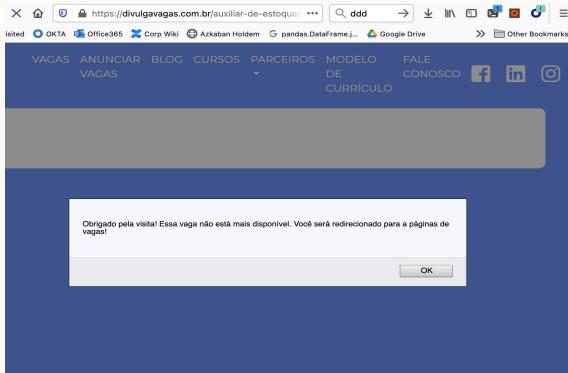
kr.jooble.org

shixiseng.com

The image shows two side-by-side screenshots of Chinese job search websites. The left screenshot is from shixiseng.com, showing a job listing for a '研发实习生 (电子电气方向)' (Research Intern (Electrical/Electronic Direction)) at KUKA. The right screenshot is from 实习招聘网, also showing a job listing for a '英语客服实习生' (English Customer Service Intern) at KUKA. Both pages include standard job details like location, salary, and duration, along with company profiles and application buttons.

The image shows two side-by-side screenshots of international job search websites. The left screenshot is from findjob.co.kr, displaying a list of Korean job postings. The right screenshot is from kr.jooble.org, showing a list of jobs for '달성군 전체 서비스 구직' (Job Vacancies in Dalseong County). Both platforms allow users to filter and sort through various job categories and locations.

The image shows two side-by-side screenshots of international job search websites. The left screenshot is from joblift.fr, showing a job listing for a 'Conseiller immobilier indépendant H/F - La Grande-Motte (34)' at 'Réseau EV Immobilier'. The right screenshot is from joblift.com, showing a similar listing for the same position. Both pages provide detailed job descriptions, company information, and application links.



Alert message

Sorry, this job is not available in your region
Search for similar jobs in your region

Redirect or not, “not available”

404 error, redirect or not,
“cannot find”

Diese Anzeige ist nicht mehr verfügbar

Single job vs. job list, “no longer”

Redirect or not, “cannot find”

Average salary
\$49,725 / Year
Based on 2,058,848 salaries
\$49,725
Low \$25,350 High \$84,533

Redirect or not, “not available”

Solutions

We should consider an investment of a hybrid approach.

Job title matching

Build a static domain knowledge base and perform job title matching against crawled content.

- Pros: Easy to understand
- Cons:
 - Job quality issues of the static domain knowledge base
 - Job title mismatches due to format difference

RegEx Rule-based

Use manually extracted rules to achieve high precision

- Pros:
 - High precision
 - Easy to understand/deploy directly in the remote scrapers
- Cons:
 - Significant manual effort needed
 - Less adaptive to changes
 - Hard to cover the long tail

AI-based model

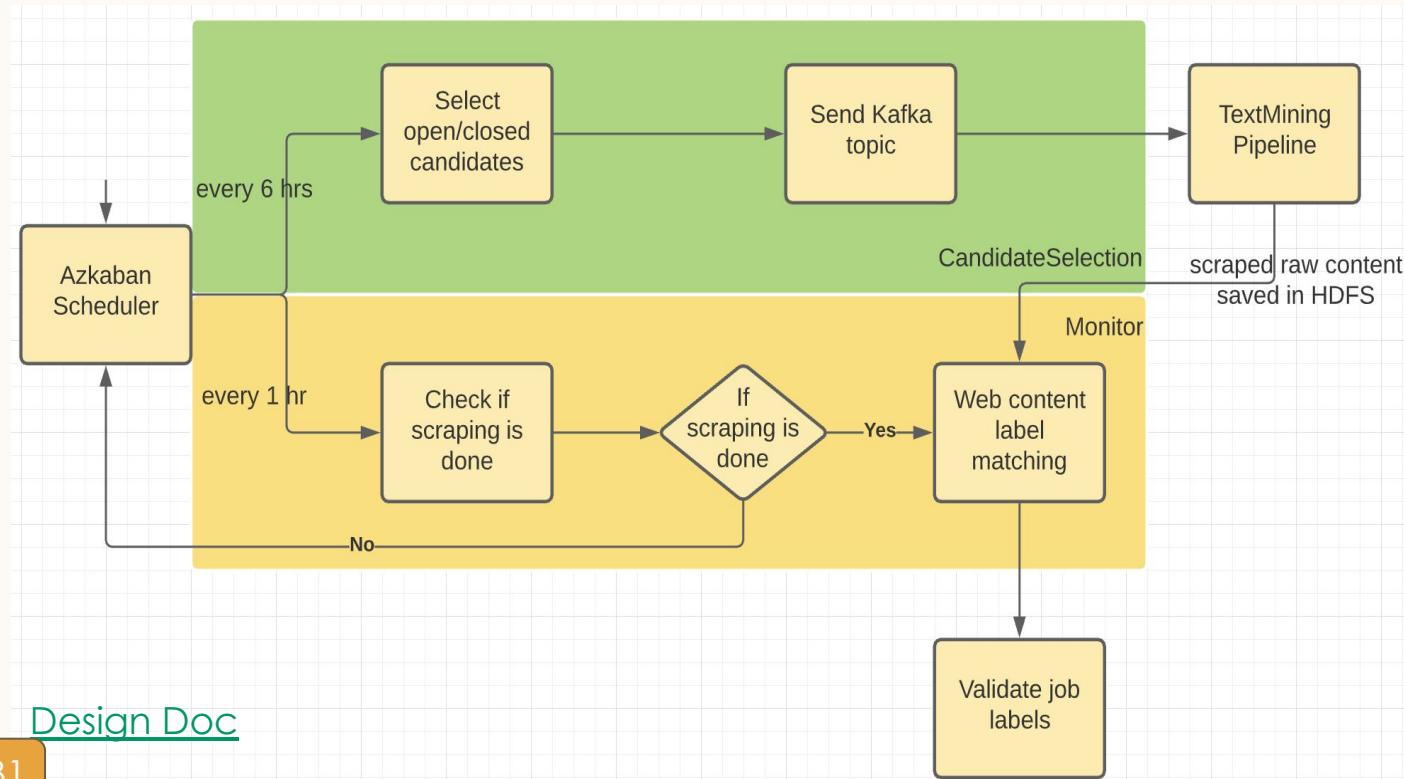
Use AI model to learn hidden rules from the raw data

- Pros:
 - Less manual effort to update and maintain the rules
 - Easy update
- Cons:
 - Hard to achieve high precision
 - Large and precise training dataset needed
 - Deployment constraints



Data Preparation Iteration 1

Data Preparation -- Iteration 1



Design Doc

Rb 2667131

Iteration 1 Blocked & Learning from Iteration 1

Why blocked?

Since the design in iteration 1 depends on the Text Mining pipeline to automatically scrape and save raw content in HDFS, it was blocked when the Text Mining pipeline wasn't launched on time.

Experience gained

- Program in Scala, Presto SQL
- General idea on how to select open/closed candidates for training data
→ **reuse in Iteration 2**
- Basic knowledge on LinkedIn internal tools

Data Preparation Iteration 2

Initial query rb 2675696.

Thank you Songtao for providing detailed feedback and scraped web content.

Data Preparation -- Iteration 2

Candidate Selecting Heuristic

Open jobs = non-JA jobs created in last 2 days & never flagged

Closed jobs = jobs closed by job ingestion in last 90-30 days & closed in raw jobs & existed in job posting + flagged closed job in recent 90 days

**Additional filters
minimize 3m+ noises**

How many jobs we select?



58.2M

Closed jobs



2.1M

Open jobs

How about sampling strategy?

Things to be aware of when sampling

For high coverage and representation, we should ensure...

- Each domain in the training data has enough samples representing both open and closed jobs.
- Acceptable amount of samples from each domain to avoid dominance.
- Cover representative domains.

Sampling methods

Based on proportion

- The amount of samples selected from every domain is based on its contribution to all jobs. Round up if needed.

Ex. We want 60k open/closed jobs.

Suppose domain D has d jobs and k jobs in total, we should select $60k \times d / k$ for D .

- The amount of samples selected from each domain is the same. Calculating performance score based on proportion (weight). Round up in the end if needed.

Ex. We select 1 job from every domain. We times the weight (d/k) when calculating score.

From top domains

Total 85,867 domains covered by 27,593,114 non-JA sourced jobs created in the past 1 month

<u>Top domains</u>	<u>Job coverage</u>
1630	90%
4357	95%
20469	99%



Stats

sample size	pre_label	domains	jobs	Total jobs
Top 2k domains, max 100 samples/domain-label	closed	1,951	189,984	339,720
	open	1,922	149,736	
Top 5k domains, max 100 samples/domain-label	closed	4,210	404,884	598,462
	open	4,062	193,578	
Top 2k domains, max 50 samples/domain-label	closed	1,951	95,615	177,985
	open	1,922	82,370	
Top 5k domains, max 50 samples/domain-label	closed	4,210	205,124	333,241
	open	4,062	128,117	

Sample Size: top **2k** domains, max **100** samples/domain-label

Feature Extraction

URL query params

The screenshot shows a job search result for a Brunel job. The URL in the browser's address bar is highlighted with a red box and contains the following query parameters: `?popunder=true&utm_source=link`. The page displays the job title "Ingenieur Technischer Vertrieb [w/m/d] [Ingenieur/in - Elektrotechnik]" and other job details like company name and location.

Company
Job location

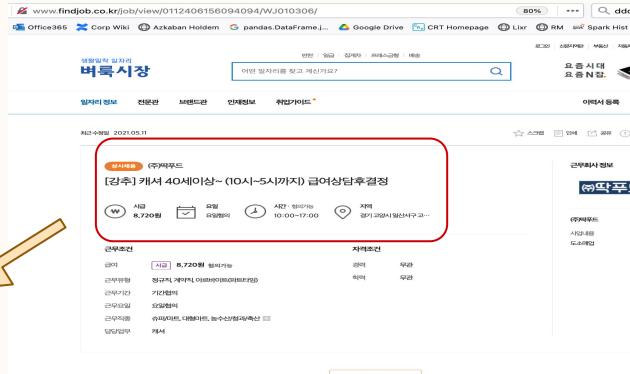
Job title
Job created date
Job type (full/part time)
Job description

Language

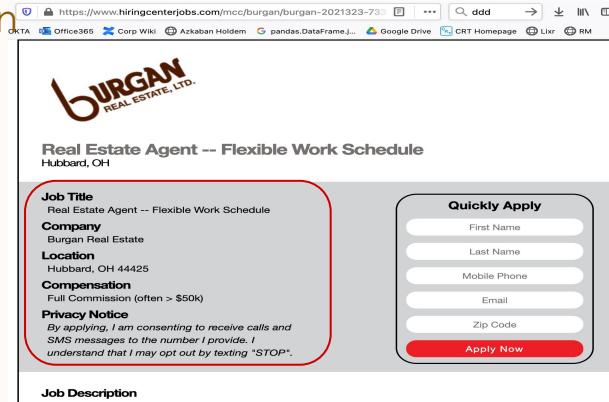
The screenshot shows a job search result for a Pflegefachkraft (w/m) position. A message box states "Diese Anzeige ist nicht mehr verfügbar". Below it, a section titled "PASSEND ZU DiesEM ERGEBNIS" lists two new jobs: "Pflegefachkräfte (m/w/div)" and "Wohnbereichsleitung (m/w/div)". Both ads include logos for UNION HILFSWERK and have a "Zum Job" button.

Embedded links

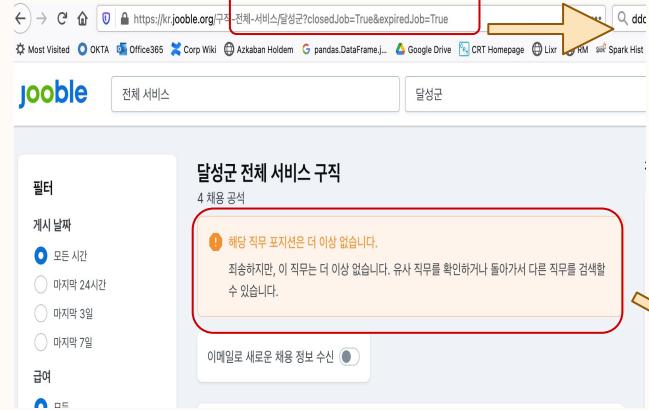
Keywords “no longer”



Job title
Job created date
Job type (full/part time)
Job description
Company
Salary
etc.

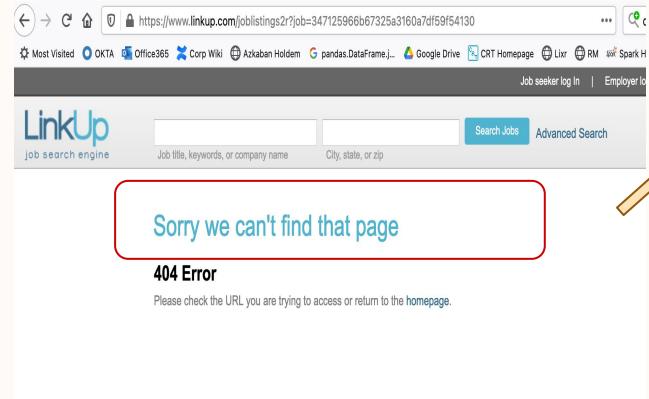


Open job



달성

URL params
expiredJob=True
closedJob=True



Sorry we can't find that page

404 Error

Please check the URL you are trying to access or return to the [homepage](#).

Closed job

Potential features we can explore

Web features

- Seed domain/final URL
domain(redirected)
- DOM structures (tag, class
names)
- URL query parameters (keywords)

Job-web matching features

- Job title/description/location
match

Job features

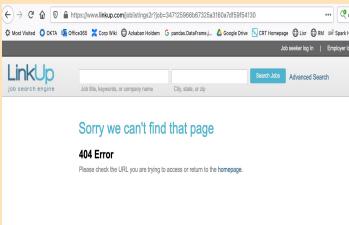
- Job engagement (clicks,
applies, favs)
- Job created date

The screenshot shows a job listing page with a header featuring a large letter 'A' and the text 'Editorial & News'. Below the header, there is a note: 'Please note this job description contains only a partial listing of duties or responsibilities that are required of the employee for this job. Duties, responsibilities and activities may change at any time with or without notice.' A section titled 'Qualifications' lists: '• Communications experience, within a design environment preferred.
• Must have minimum of 3-5 years of experience.
• Experience working with front-end technologies such as React, CSS, and JavaScript.' A large 'We are hiring!' button is centered below the qualifications. Below the button are filters for 'Job location' (All locations), 'Job category' (All categories), and 'Job type' (All types). A search bar shows 'Software Development'. A job listing for 'Senior Front-end Developer' is displayed, showing it's a Full-Time position in the US, remote friendly, with a 'View job' button.

Features chosen for modeling

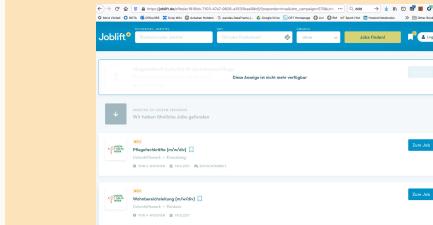
1. Length of raw html

Some closed job might only have a message saying something similar to "this job is already closed" while open jobs usually include detailed job description, responsibility, company introduction etc and thus longer.



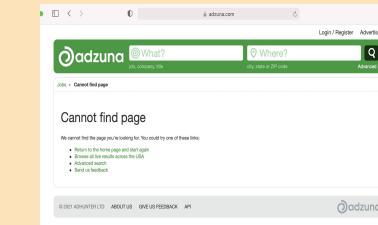
2. Totals of links

Some domains suggest other opportunities to the job seekers if the job they are currently looking at is closed. Those domains usually include a list of embedded links to other jobs.



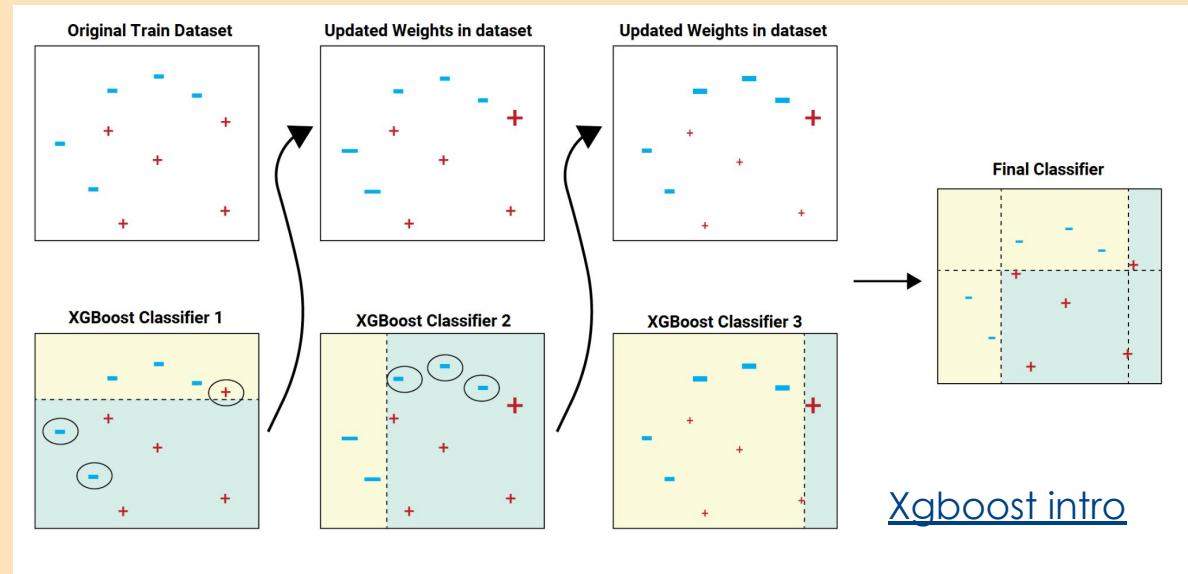
3. Job title matching

If the job title doesn't match, this job, with high probability, is closed.



Modeling & Evaluation

[DEMO notebook](#)



70.5%

Accuracy

78.2%

66%

Precision

78.3%

81%

Recall

78.5%

Before: This represents the performance of xgboost model with 2 features w/o tunings.

After: This represents the performance of xgboost model as a desired outcome of tuning.

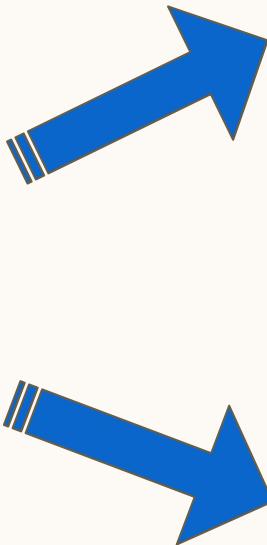


For who to
take it over,

- 1 Manage a clean training dataset
- 2 Increase sample size + Tuning
- 3 Review failure cases & enrich features
- 4 Transport to PCV2

Challenges & Takeaways

- **Everything is new.** New programming language. New tools. New knowledge.
- **Limited related background.** Learn everything by myself from the very beginning.
- **Dependency challenge.** Depending pipeline wasn't launched on time. Had to switch to another approach.



- **Technological takeaways**
 - Basic knowledge on ML modeling
 - Code/Debug in Scala, SQL, Python
 - Basic understanding on internal tools including PCV2, Frame, avro-schema, Azkaban, etc.
- **Non-technological takeaways**
 - Be chill when blocked and quickly proceed to solve problem actively
 - Analyze problems independently and support my statements with solid stats
 - Ask questions. Reach out to experienced engineers for suggestions professionally

Q & A

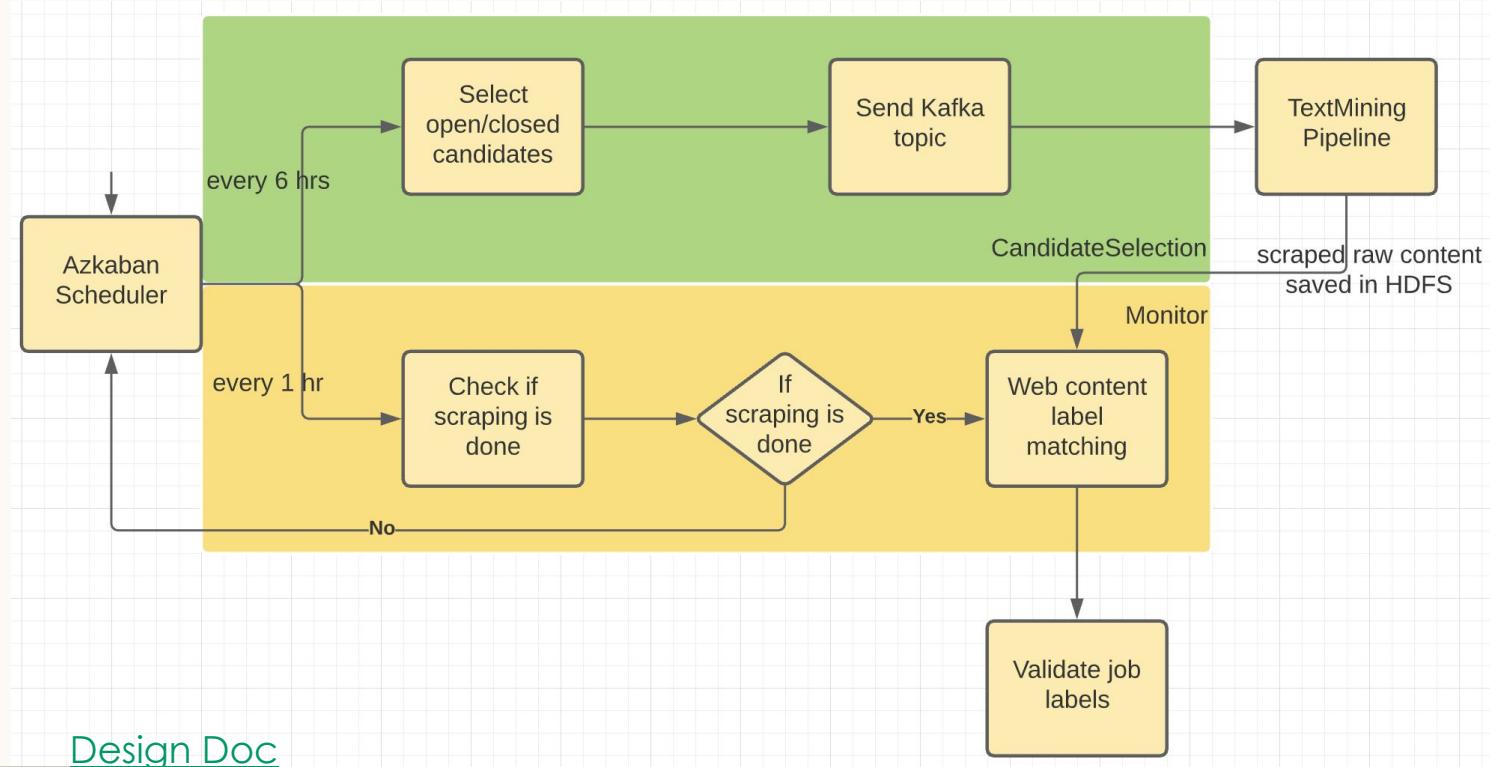
Special thanks to Songtao and Yusi



Thank you!

Appendix

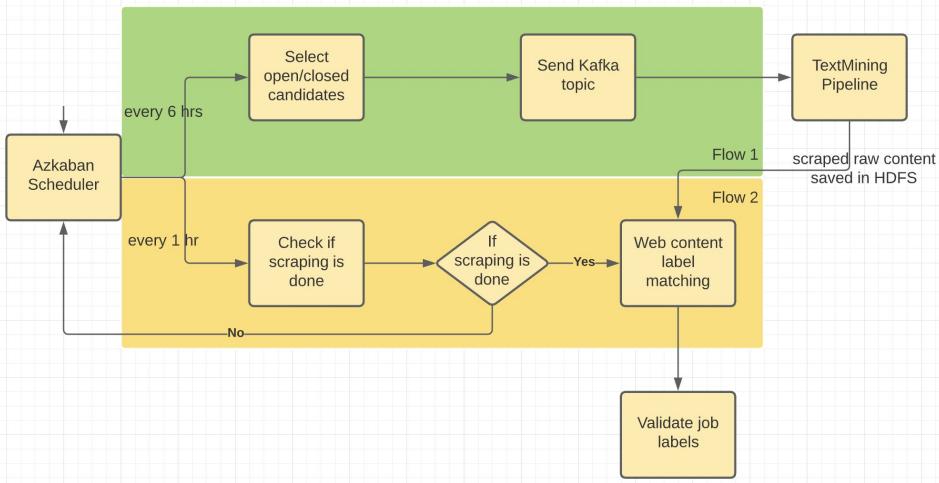
Data Preparation -- Iteration 1



Design Doc

Rb 2667131

Data Preparation -- Iteration 1



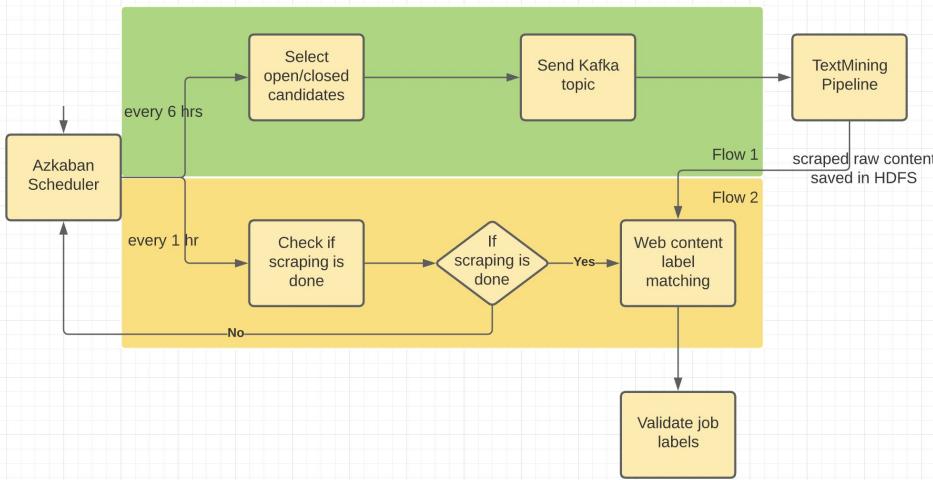
Flow 1

- Input: source, raw_jobs, jobs_column_prod, user_flagging_union, datepartition, sample size
- Output:
 - Save job_id, url and label in HDFS with batch_id as folder name
 - Send a TextMiningRequestMessage for every selected candidate

What happens next

1. Select urls(urls is labeled) -> send to Kafka topic1
2. Ubiquity-backend consumes Kafka messages, and then schedules the scraping which is done by Scraper system
3. Scraper service sends scraped web content to Kafka topic2
4. Messages in Kafka topic2 will be materialized to HDFS automatically and further processed by flow 2

Data Preparation -- Iteration 1



Flow 2

- Input: workflow dir of flow1 and flow2, TEXT_MININGJobIngestionSourceMessageV2
- Output: raw_content with labels, job_id, url, datapartition registered in a partition table

Now we should have raw html paired with its open/closed label and should be ready to proceed to feature extraction for modeling.