SkillFSA: Extracting Job Skills From Resumes Using Finite State Acceptors

Andres, John Austin T.
Saint Louis University
Maryheights, Bakakeng Sur, Baguio City, Benguet, Philippines
+63 908 7785 893
lukecabuenas@gmail.com

Capistrano, Bryan Kenneth F.
Saint Louis University
Maryheights, Bakakeng Sur, Baguio City, Benguet, Philippines
+63 949 595 5638
bryan@gmail.com

Ferrer, Miguel Angeles G.
Saint Louis University
Maryheights, Bakakeng Sur, Baguio City, Benguet, Philippines
+63 917 659 0303
linkgen42070@gmail.com

Madriaga, Jeremy M.
Saint Louis University
Maryheights, Bakakeng Sur, Baguio City, Benguet, Philippines
+63 926 665 1774
madriaga.jeremy.jm@gmail.com

1. INTRODUCTION

Employers frequently receive hundreds or even thousands of resumes in the competitive job market of today for a single job opening. In order to manually review each resume and determine each applicant's skills and qualifications, recruiters may find it time-consuming and challenging given the volume of applications [1]. Recruiters and job seekers may become frustrated as a result because qualified candidates may be passed over during the screening process.

Finite State Acceptors are a particular kind of automaton that can identify and categorize input sequences in accordance with a pre-established set of rules [2]. Employers can automate the process of reviewing resumes for specific qualifications by designing a Finite State Acceptor specifically made to identify job skills.

The time-consuming and error-prone manual screening of resumes by recruiters is the issue that this paper seeks to address. Recruiters often invest a lot of time in screening resumes only to discover that the majority of applicants are unqualified for the position. Recruiters can concentrate on the most qualified candidates and spend less time and effort identifying job skills by automating the screening process [4]. Automating the screening procedure can also aid in removing human bias from the selection process, resulting in a more impartial and objective evaluation of candidates [5].

As the demand for web developers is projected to increase significantly between 2018 and 2028 [7], it is anticipated that employers will seek effective methods to identify and classify the most suitable candidates for the role. To address this need, there may be a growing trend towards the development and use of tools and technologies that can assist employers in assessing the qualifications and skills of job applicants for web development positions.

The goal of the paper is to create a program that uses a Finite State Acceptor to automatically recognize and extract job skills from resumes, and if the extracted job skills match the requirements of a recruiter it would also get the e-mail address of the applicant. Through this program, recruiters will spend less time and effort screening and evaluating resumes, giving them more time to concentrate on more strategic tasks [3].

The paper's solution can assist recruiters in more effectively identifying qualified candidates and improve the chances of finding the best fit for a specific job opening by utilizing technology and automating a time-consuming task [6].

2. METHODOLOGY

To accomplish the study's goal, the researchers used a variety of methods, such as data-gathering, pre-processing of the data, skill set identification, finite state acceptor design, comparison with regex, and creation of proof-of-concept web application .

2.1. Data Collection

It is necessary to gather a dataset of resumes to serve as the program's starting point in order to develop and test a program for job skills extraction from resumes.

Data for the study was sourced by the researchers from freely accessible websites like kaggle.

2.2. Pre-processing

Since raw dataset/s may contain a lot of noise, such as special characters, multiple new lines and/or whitespaces, the researchers preprocessed it to remove such noise, stop words, short words. To obtain consistency, the words are converted into lowercase and are tokenized into a list.

2.3. Skill Set Identification

Given the focus of the study on web development, it is anticipated that all resumes analyzed will exhibit proficiency in basic web development skills. Specifically, note that every employer would require a different set of skills. As such the research paper will be focusing on a source based on a survey of the top web developer skills required by employers.

2.4. Finite State Acceptor Design

This involves designing and developing a Finite State Acceptor specifically tailored to the task of job skills extraction from resumes. This involves identifying the key job skills and qualifications that employers are looking for in candidates, and developing a set of rules or patterns that the Acceptor can use to identify these skills in resumes.

2.5. Comparison with Verified Checkers

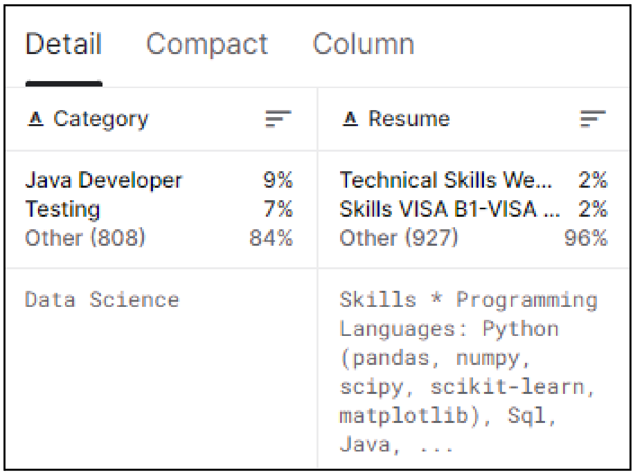
For the researchers to understand whether the FSA is operating as intended. Verified checkers will be used alongside the FSA and the dataset.

2.6. Creation of Proof-of-Concept

A sample proof of concept application will be developed to put the researchers' newfound knowledge to use. On the basis of the earlier methodologies, the requirements for the application may be determined.

3. RESULTS AND DISCUSSION

3.1 Data Collection



Category		Resume	
Java Developer	9%	Technical Skills We...	2%
Testing	7%	Skills VISA B1-VISA ...	2%
Other (808)	84%	Other (927)	96%
Data Science		Skills * Programming Languages: Python (pandas, numpy, scipy, scikit-learn, matplotlib), Sql, Java, ...	

The study took into account Gaurav Dutta's resume dataset from Kaggle. There are 962 rows and two columns, titled Category and Resume. Only the Resume column will be used by the researchers for the analysis and testing of the FSA.

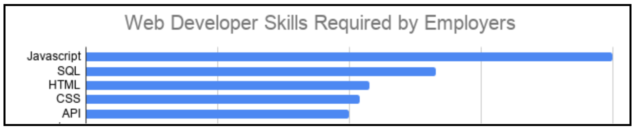
3.2 Pre-processing

Example of a noisy text:
"Python (pandas, numpy, scipy, scikit-learn, matplotlib), Sql, Java, JavaScript/JQuery. *"

Tokenized cleaned text:
["python", "panda", "numpy", "scipy", "scikitlearn", "matplotlib", "sql", "java", "javascript", "jquery"]

For the FSA to function properly, the pre-processing step is essential because it prevents noise from being discovered, such as a string of supposedly accepted javascript that is discovered in a string containing a special character, such as "javascript2," which would cause it to be rejected.

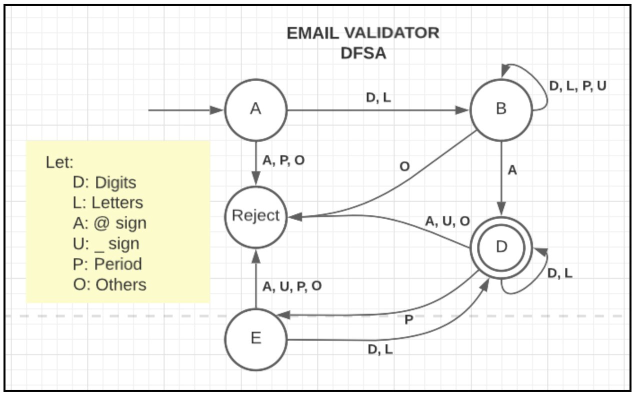
3.3 Skill Set Identification



The researchers used data from BeamJobs' website to gather information for the skill set analysis. Due to its thorough coverage of web developer job descriptions from 130 different companies, this particular source was chosen. Based on the findings, the researchers chose the top 4 skills as their skill set for the finite state acceptor and application.

3.4 Finite State Acceptor Design

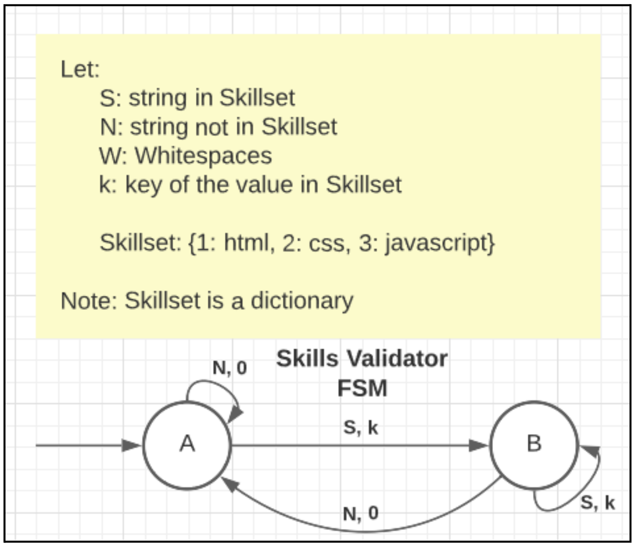
The researchers made Deterministic Finite State Acceptor (DFSA) model for Email Validation. These models were used to represent and analyze systems that exhibit a discrete sequence of states and transitions between those states.



The DFSA model for Email Validator represents 5 states where:

- State A (Initial State) transitions to State B if “Digits or Letters” were detected, else State A will transition to Reject State if “@, . , Others” were rejected.
- State B transitions to State D (Accepting State) if “@” sign was detected, else if “Digits, Letters, Period, and _” were detected, It will loop to itself (State B), else State B will transition to Reject State if “Others” were detected.
- State D or the “Accepting State” transitions to State E if “Period” was detected, else if “@, _, Others” were detected, it will transition to Reject State. Else, State D will loop to itself if “Digits or Numbers” were detected.
- State E transitions to State D (Accepting State) if “Digits or Letters” were detected, else State E transitions to Reject State if “@, _, . , Others” were detected.

While the Skills Validator FSM consists of only 2 states, State A and State B. This FSM checks every word as an entry from the resume.

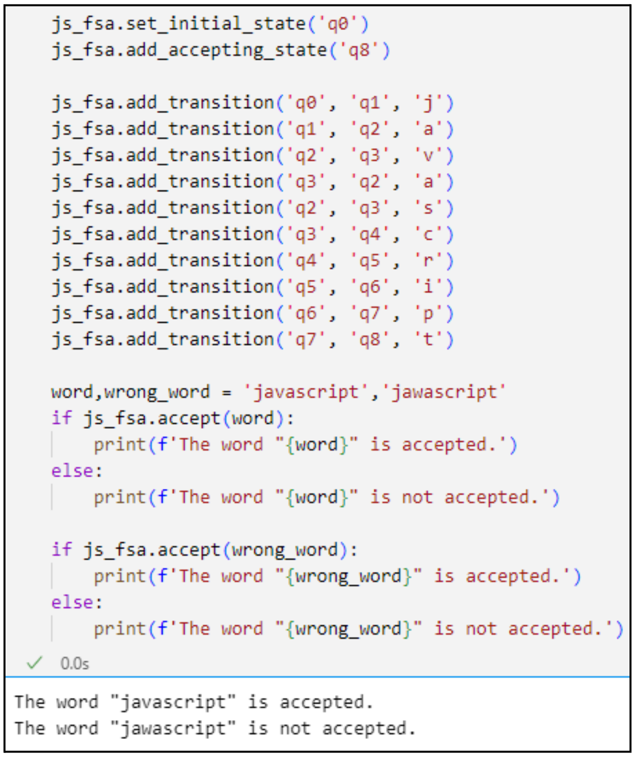

Let:
S: string in Skillset
N: string not in Skillset
W: Whitespaces
k: key of the value in Skillset
Skillset: {1: html, 2: css, 3: javascript}
Note: Skillset is a dictionary
Skills Validator
FSM
N, 0
A
S, k
B
N, 0
S, k

The DFSA model for Skillset Validator represents 2 States where:

- State A (Initial State) transitions to State B if “string in Skillset” or a value in Skillset dictionary were detected, else State A will loop to itself if “String not in Skillset or 0” were detected.
- State B transitions to State A if “string in Skillset or 0” were detected. While it loops to itself if it detects another “skill” string word from the data.
- The outcome of the implemented Finite State Machine (FSM) is a generated string that may potentially include a combination

of the digits 0, 1, 2, and 3. Subsequently, the web application undertakes a verification process to determine the presence of the numbers 1, 2, and 3 within the string. This validation serves as an indication of the inclusion of the skills HTML, CSS, and JavaScript in the analyzed resume, respectively.

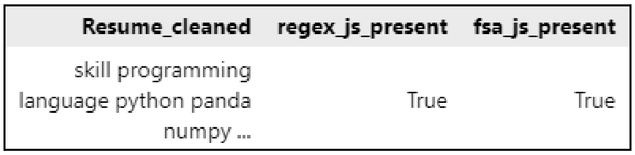
2.5. Comparison with Regex



```
js_fsa.set_initial_state('q0')
js_fsa.add_accepting_state('q8')

js_fsa.add_transition('q0', 'q1', 'j')
js_fsa.add_transition('q1', 'q2', 'a')
js_fsa.add_transition('q2', 'q3', 'v')
js_fsa.add_transition('q3', 'q2', 'a')
js_fsa.add_transition('q2', 'q3', 's')
js_fsa.add_transition('q3', 'q4', 'c')
js_fsa.add_transition('q4', 'q5', 'r')
js_fsa.add_transition('q5', 'q6', 'i')
js_fsa.add_transition('q6', 'q7', 'p')
js_fsa.add_transition('q7', 'q8', 't')

word,wrong_word = 'javascript','jawascript'
if js_fsa.accept(word):
    print(f'The word "{word}" is accepted.')
else:
    print(f'The word "{word}" is not accepted.')

if js_fsa.accept(wrong_word):
    print(f'The word "{wrong_word}" is accepted.')
else:
    print(f'The word "{wrong_word}" is not accepted.')
✓ 0.0s
```

```
The word "javascript" is accepted.
The word "jawascript" is not accepted.
```

A sample of creating the FSA is given. Keep in mind that the researchers' FiniteStateAcceptor class contains the set_initial_state, add_accepting_state, add_transition, and accept methods.



Resume_cleaned	regex_js_present	fsa_js_present
skill programming language python panda numpy ...	True	True

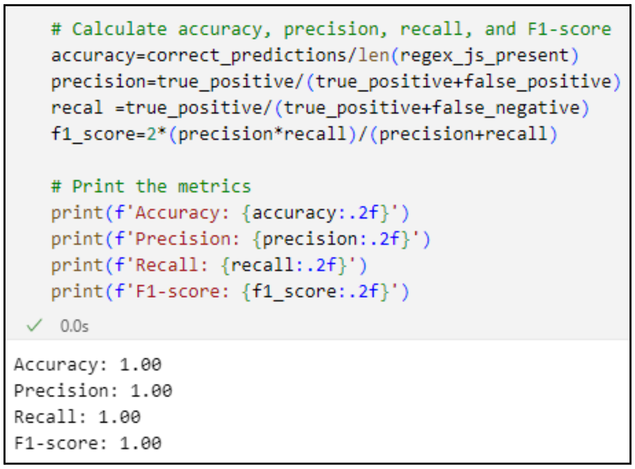
Utilizing the dataset and the Python "re" which is a regular expression library. The Pandas library is used to add an additional 6 columns. Three columns are used for regex ("javascript," "html," and "css") and three are used for FSAs ("javascript," "html," and "css").

The outcomes of applying regex and the FSA are provided. The resume's initial text is as follows:

```
Programming Languages: Python (pandas, numpy, scipy, scikit-learn, matplotlib), Sql, Java, JavaScript/JQuery.
```

As can be seen, the resume includes JavaScript.

The following metrics are also provided for javascript:

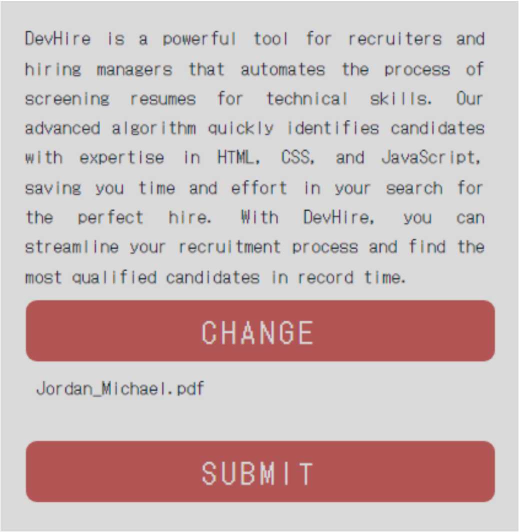


```
# Calculate accuracy, precision, recall, and F1-score
accuracy=correct_predictions/len(regex_js_present)
precision=true_positive/(true_positive+false_positive)
recal =true_positive/(true_positive+false_negative)
f1_score=2*(precision*recall)/(precision+recall)

# Print the metrics
print(f'Accuracy: {accuracy:.2f}')
print(f'Precision: {precision:.2f}')
print(f'Recall: {recall:.2f}')
print(f'F1-score: {f1_score:.2f}')
✓ 0.0s
```

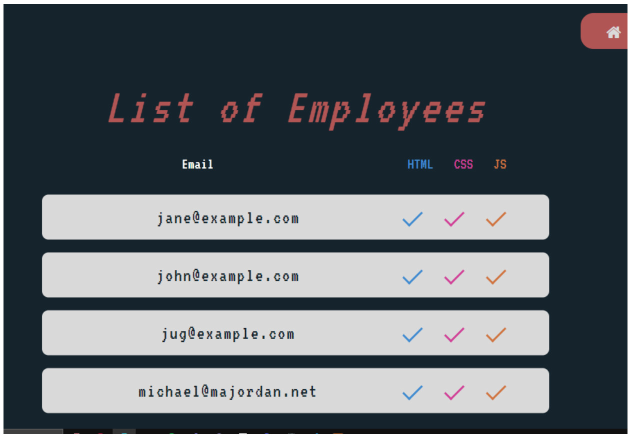
```
Accuracy: 1.00
Precision: 1.00
Recall: 1.00
F1-score: 1.00
```

As can be seen from the metrics for javascript, the FSA is on par with the library.

3.6. Creation of Proof-of-Concept



DevHire is a powerful tool for recruiters and hiring managers that automates the process of screening resumes for technical skills. Our advanced algorithm quickly identifies candidates with expertise in HTML, CSS, and JavaScript, saving you time and effort in your search for the perfect hire. With DevHire, you can streamline your recruitment process and find the most qualified candidates in record time.



List of Employees

Email	HTML	CSS	JS
jane@example.com	✓	✓	✓
john@example.com	✓	✓	✓
jug@example.com	✓	✓	✓
michael@majordan.net	✓	✓	✓

The proof of concept application is a web application with a front end written in simple javascript, html, and css. The back end consists of routing using the flask library. The PyPDF2 library is used to extract the resume's contents, which are then preprocessed for consistency. Finally, the server applies the FSAs the researchers has programmed to determine whether the resume contains the necessary skills. The applicant's email is then scraped after the three skills are seen.

4. CONCLUSION

4.1 Conclusion

The researchers have created a web application that utilizes a non-deterministic finite state acceptor that checks for the email address of potential employees. The displayed tool will allow the users or employers to maximize productivity and efficiency in the employment process.

4.2 Future Directions

For the future directions of this study, future researchers may use the tool for further academic pursuit. Different pathways may open opportunities for future researchers such as:

- Incorporation of other areas of expertise into the tool to enhance its effectiveness.
- Improvement of the web application design to allow users to access better information.
- Implementation of a point system to rank potential candidates for more accurate employment recommendations.
- Use of additional resume data beyond just skill sets to generate more varied results.
- Deployment of the web application to be used in real-world employment processes.

5. REFERENCES

[1] Ideal. (n.d.). Resume Screening: The Complete Guide for HR Professionals. Retrieved from https://ideal.com/resume-screening/

[2] Automata Theory. (n.d.). In Encyclopædia Britannica. Retrieved from https://www.britannica.com/topic/automata-theory/Classification-of-automata

[3] Tengai. (2021, February 23). How AI Recruiting Solutions Help Reduce Bias in Your Hiring Process. Retrieved from https://tengai.io/blog/how-ai-recruiting-solutions-help-reduce-bias-in-your-hiring-process

[4] Infinity Staffing & Training Solutions. (n.d.). Resume Screening: A How-To Guide for Recruiters. Retrieved from https://www.linkedin.com/pulse/resume-screening-how-to-guide-recruiters-infinitysts

[5] Forbes Human Resources Council. (2021, October 14). Understanding Bias in AI-Enabled Hiring. Forbes. Retrieved from https://www.forbes.com/sites/forbeshumanresourcescouncil/2021/10/14/understanding-bias-in-ai-enabled-hiring/?sh=728ae0847b9

[6] Kelly, J. (2023, March 15). How AI-Powered Tech Can Help Recruiters and Hiring Managers Find Candidates Quicker and More Efficiently. Forbes. Retrieved from https://www.forbes.com/sites/jackkelly/2023/03/15/how-ai-powered-tech-can-help-recruiters-and-hiring-managers-find-candidates-quicker-and-more-efficiently/?sh=5cdb9a853a3f

[7] BeamJobs. (n.d.). Web Developer Skills. Retrieved from

https://www.beamjobs.com/resume-help/web-developer-skills

[Dataset] GauravDuttaKIIT. (2021). Resume Dataset. Kaggle. Retrieved from https://www.kaggle.com/datasets/gauravduttakiit/resume-dataset