1 Shane Mulligan; BSc (Computer Science)

specialisation scripting, domain specific languages, natural language engineering

1.1 Cover Letter

To my prospective employer,

Please consider me for this role.

Here is a terminal recording of using a script I made with bash and python to create a wordcloud of all the python libraries I've installed and tinkered around with over the last year.

`https://mullikine.github.io/posts/python-library-wordcloud/`

My dream job is a Language Engineer; essentially, creating DSLs to interface with Natural Languages. I have been in the workforce as a software developer for about 10 years. I have been programming for around 20. I am an expert at scripting languages, particularly python and proficient at many other languages, including Haskell. I am experienced with microservices; I've used AWS, Kubernetes and Google Cloud. I am a scientific programmer; I can research and create quality documentation. I have been building my own learning environment with unix tools since 2010. I have my BSc in Computer Science. I also have experience in creating and debugging automated tests for "Hardware In the Loop". I have work experience and open source experience in Golang, C++, and emacs lisp. My style is language agnostic; I easily pick up new languages, file formats and tooling. I build and integrate AI-assisted programming tools (See the link below). I'm a very strong emacs user and I have a blog where I talk about plugins I have made and the things I learn about. Earlier this year I automated a pipeline to scale a platform for automating pull requests to GitHub. I have been using Ubuntu Linux for the last 10 years; I am as knowledgeable as they come with regard to dealing with linux. I am, however setting my sights on NixOS and have started migrating. I studied Information Retrieval at postgrad level and have built my own code search engine. I've also built tools for interfacing with ElasticSearch. I am very capable and hard working.

Thank you for your consideration,

Shane Mulligan

1.2 Motivation

I must continue abstracting my environment to integrate more things into it. This means buliding programming languages and taking advantage of Language Models (NLP).

1.3 Links

- http://mullikine.github.io - Daily Blog Please look at my blog. It's my real CV.
- GitHub, gists

1.4 Education

Degree	Field	Insitution	Status
BSc	Computer Science	University of Otago	Finished 2010
PGDip	Information Retrieval	University of Otago	Started 2018

1.4.1 Highlights

- Won the COSC343 robot competition
- Completed a GitHub search engine using GHTorrent and BigQuery.

1.4.2 Interest papers

- Health science 1st year
- Chemistry (200 level)

- Anatomy (200 level)
- Information Retrieval (400 level)
- Neural Networks (400 level)

1.4.3 Books read

Coherence in natural language. Data structures and applications

1.5 Volunteer work

1.5.1 School of Computer Science

Ongoing. Tutoring junior school and high school students.

started 2018

1. proud moments
   (a) portfolio gallery (circa. 2018) Guided 2 high school students in building an interactive portfolio gallery for the Otago Settler's Museum.
      - Taught CSS, javascript, php.

   (b) built a bot that can play the board game codenames (circa. 2018)
      - Taught the concept of 'word vectors'.

1.6 Work experience

1.6.1 CodeLingo Ltd

job title Software Engineer (contractor)

Development / automation

Golang, bash, kubernetes

1. Examples of pull requests generated and made automatically

   These pull requests were generated by the automated CodeLingo platform.

   `https://gist.github.com/1860bfea2a9e1e3b3bbb96b95a11bdd0`

   started Sep 2018
   ended May 2019

2. Languages used

language	context
golang	Built the Gometalinter lexicon
shell	Pipeline
python	GitHub API
ElasicSearch Lucene	Debugging
BigQuery standard SQL	GHTorrent / Bigquery for Github
JavaScript	Unit testing
emacs lisp	Building an environment for a new language (CLQL)

3. Highlights
   - 0 to 1000 automated github pull requests over 8 months.
   - Pipeline outreach scaled the platform from 10 to 300 app installs.

1.6.2 Crown Equipment Corporation

job title HIL (hardware-in-the-loop) Test Engineer

Continuous integration.

Embedded c/c++.

started Jan 2015

ended Sep 2017 (2 years 8 months)

1. primary languages used
   - C++13
   - python
2. Responsibilities
   - HIL (hardware-in-the-loop) rigs / integration tests
   - Implement driver for 3D Basler Camera
3. Highlights
   - Built a platform for correlating error messages with logs and code using the the Sphinx open source search engine.

1.6.3 Tracmap

job title Software Engineer

Embedded programming

Full-stack web development

started 2011

ended 2013 (2 Years 2 months)

1. primary languages used
   - C++
   - python
   - javascript
   - postgresql
2. Highlights
   - Ported the firmware from the older TM4 head units to the then prototype TM5 headunit.
3. Images
   - 2012: TM4 needs love
   - 2019: First TM5 returns home

1.7 Open-source Projects

1.7.1 Age of Kings Trigger Studio

`http://aok.heavengames.com/blacksmith/showfile.php?fileid=12103`

The most popular Age of Empires II scenario editor. Downloaded 10,000 times. Used in making campaigns for Age of Empires II HD edition and expansions.

Software	Purpose
IDA Pro	Reverse engineering
Visual Studio 2005	Compiling
emacs/vim	Programming

1.8 Skill set

- Language agnostic / polyglot
- Prefers autonomy/ self-management
- Continuous learning
- I do everything inside a terminal (and I make it look good)
- Prefers automation in almost every situation

1.8.1 General knowledge / skills (context)

Area	Context
Build tool automation	Many build tools
Polyglot programming	Many command line tools and programming languages
Scripting / pipelines / automation	CI, bash, jq, python, haskell, expect
Building terminal user interfaces	golang, semantic highlighting, real-time feedback, fzf, emacs
Building debugging tools	tracing, automated git bisect, extending magit
Metaprogramming	code generation, emacs lisp, racket
Preprocessing	rosie lang, pcre, sed, awk
Algorithms	Information retrieval / NLP
Information retrieval	Google automation, bigquery, sql generation
Documentation	report/blog writing, latex, org-mode
Web development	html, css, ajax, progressive enhancement
Deep learning	word embeddings, Keras, pytorch, 'GPT-2/the transformer'
Continuous Integration	Jenkins, JenkinsX, CodeLingo, CloudBees
Natural language processing	Spacy, word vectors <-> words, BERT (transfer learning)
Functional programming	racket, haskell
Telco / Microservices / Kubernetes	logs, searching, automation, debugging, scripting
Bayes	scripting with problog
Building editing environment	building emacs modes, fixing bugs in emacs
Research / learning	automated Arxiv search, hacker news search, presentation

1.8.2 General knowledge / skills (evidence)

Area	Evidence from my blog
Build tool automation	Automating build systems for many languages
Polyglot programming	Languages supported by my development environment
Scripting / pipelines / automation	Reading YouTube rather than watching it
Building terminal user interfaces	Complex Dwarf Fortress macros with tcl/expect, emacs and tmux
Building debugging tools	tooling TensorFlow Debugger (tfdb) and emacs DAP mode for emacs
Metaprogramming	Practical macros in Racket C++ template metaprogramming with Racket
Preprocessing	Filtering text streams
Algorithms	The Illustrated Transformer `https://mullikine.github.io/glossary.html`
Information retrieval	Tremendous Task: Searching for code on GitHub with BigQuery and GHTorrent
Documentation	Graphviz and Hugo Entropy, Cross-Entropy and KL-Divergence
Web development	The Semantic Web and Ontology
Deep learning	The Illustrated Transformer
Continuous Integration	`http://codelingo.io` Overview of modern Continuous Integration tools
Natural language processing	Named Entity Recognition
Functional programming	GHCi and Haskell code intermixed in babel
Telco / Microservices / Kubernetes	telco github GCP
Bayes	(WIP) Probabilistic programming with problog
Building editing environment	Browsing sqlite3 databases with edbi for emacs
Research / learning	Review of 'Language Engineering; Harnessing the Power of Language (2004)'

I am always learning:
`https://mullikine.github.io/glossary.html`

1.8.3 Tools

tool / skill
emacs
vim
Deep TabNine
GPT-2
ctags
Google search automation
Code generation
Code snippet search

1.8.4 Programming languages

I support many languages in my environment.
`https://mullikine.github.io/posts/emacs-languages-supported/`

—

Language strengths

Table 1: legend

key	
OOP	object-oriented
FP	functional programming
Exp.	experienced

Language	Strong	Exp.	Advanced skills
Python	yes	yes	code-gen, reflection, own library, OOP, FP
bash / zsh	yes	yes	code-gen, own library, FP
c	yes	yes	code-gen
c++ (98)		yes	code-gen
c++ (13)		yes	code-gen
SQL	yes	yes	code-gen
Go	yes	yes	own library
CSS	yes	yes	code-gen
Haskell		yes	reflection, own library, FP
common lisp	yes		code-gen, FP
tcl/expect	yes	yes	code-gen, own library
emacs lisp	yes	yes	code-gen, own library, metaprogramming
scheme / racket	yes	yes	code-gen, own library, metaprogramming
problog		yes	code-gen
perl	yes	yes	
sed, PCRE	yes	yes	code-gen
awk	yes	yes	code-gen, own library
clojure			
javascript		yes	
java		yes	
jq	yes	yes	code-gen
graphviz	yes	yes	code-gen
latex	yes	yes	code-gen
vimscript	yes	yes	code-gen, own library
rosie			
CodeLingo Query Language	yes	yes	code-gen, own library, metaprogramming
prolog			code-gen
rust			
typescript			
scala			
smalltalk			
R			

1.9 References

1.9.1 Dr Zhiyi Huang

email zhuang@cs.otago.ac.nz

Associate Professor
Department of Computer Science
University of Otago
Dunedin, New Zealand

1.9.2 Jesse Meek

email waigani@gmail.com

CEO
CodeLingo
Dunedin, New Zealand

1.10 Contact details

phone +64 3 4777 071

mobile +64 21 146 2759

mobile +64 22 589 5536

email mullikine@gmail.com

1.10.1 Linkedin

www.linkedin.com/in/shane-mulligan-811b942b/