Labour dynamics in the age of automation: detecting emergent skills in labour markets from job ads description

Yaozhong Liu

A thesis submitted for the degree of
Bachelor of Advanced Computing (Honours)
The Australian National University

February 2021

Except where otherwise indicated, this thesis is my own original work.

Yaozhong Liu
Wednesday 3rd February, 2021

To my parents, supervisors, friends, classmates, and all the people who supported me by my back all the time.

Acknowledgments

I believe that for most people, 2020 is not a pleasant year among the years they have experienced in their lives. Because of the entry restrictions, I was locked down in China and can not go back to Australia to start my final year in ANU locally. Fortunately, I successfully finished my honors project and achieved good results in different courses with the help of many people. I would like to express my gratitude to all the people who helped all the way around this extraordinary year.

I would say thank you to my parents who rally behind, giving me unconditional support.

Thanks to my friends as well as my classmates. Despite some of them are still in China while some of them are in Australia, we encourage and support each other to keep optimistic to overcome difficulties.

Thanks to everybody in **Behavioral Data Science** group. They are always willing to help when I met problems or ask them question about daily life or academia. Thank my senior Quyu Kong for offering aid with patience when I need tutorial to solve problems and when I need advice for a better presentation.

I would like to give my biggest thanks to Assistant professor Marian-Andrei Rizoiu my supervisor, and Mr Nik Dawson. I received comprehensive guidance from them and was inspired by their punctilious attitude towards academic knowledge. The words fail to express my heartfelt thanks to their kindness and patience since they not only give me instruction in this research, but they also teach me what is good researchers and how to become one of them.

I should be grateful for this special experience. Every people I met, everything I go through will influence me in a imperceptible way and make me better. I will carry all of these and begin my next journey.

Abstract

With the prosperity of the online labour market, more and more employers are willing to post recruitment advertisements on the websites. The demand of the labour market changes in a unpredictable speed and many new skills emerge then quickly reflect on the labour market. For the purpose of extracting the existing skills and even find new emerging skills, we leverage the popular natural language processing (NLP) framework, and build reliable model to accomplish this goal.

In this research, we use job advertisements from five different English speaking countries including Australia, Canada, UK, Singapore, US and mainly use a natural language processing frame work called spaCy to build named entity recognition (NER) model to identify the skills in these job ads. We propose two kinds of model one is called proprietary model, the other is called joint model to this skills detecting task. We summarise the pros and cons for these two kinds of model respectively and give suggestions about what kind of model should be chosen to solve different problems. Additional to the spacy NER model, we also analyze the factors leading to wrong prediction and compare with the models trained by another NLP framework Flair.

The contributions of our research consists of two aspects. One is that we built a reliable NER model whose prediction result is impressive using spaCy and we prove that spaCy is a more suitable choice than Flair when you have rigid time limitations for training, relatively high requirement for accuracy and does not demand powerful device. Furthermore, no prior work provides a NER model to finish this task, our work can play an important role in future related research not necessarily restricted in extracting skills from job ads.

Keywords: Natural language processing (NLP), Named entity recognition (NER), spaCy, Flair, embeddings, index annotation, BIO annotation, proprietary model, joint model.

Contents

Acknowledgments **vii**

Abstract **ix**

1 Introduction **1**
1.1 Thesis Statement . . . 1
1.2 Introduction . . . 1
1.3 Thesis Outline . . . 2

2 Background and Related Work **5**
2.1 Motivation . . . 5
2.2 Related work . . . 5
2.2.1 Natural Language Processing . . . 6
2.2.2 Named entity recognition . . . 6
2.2.3 Popular nlp tools . . . 7
2.3 Summary . . . 9

3 Data processing and synthesis **11**
3.1 Data Process . . . 11
3.2 Summary . . . 14

4 Method **15**
4.1 Data acquiring and evaluation . . . 15
4.2 Framework comparison and selection . . . 16
4.2.1 Creating spaCy training set and spaCy model training . . . 16
4.2.2 Creating Flair training set and Flair model training . . . 20
4.2.3 Summary of spaCy and Flair . . . 21
4.3 Model training . . . 22
4.3.1 Proprietary model . . . 22
4.3.2 Joint model . . . 22
4.4 Evaluation methods . . . 23
4.5 Summary of this chapter . . . 26

5 Results **29**
5.1 Model performance . . . 31
5.1.1 Validation of our training strategy . . . 31
5.1.2 Comparison of proprietary model and joint model . . . 33

5.1.3 The factor lead to erroneous predication and the performance gap between two kinds of models 37
5.1.4 Other characteristics of spaCy NER model in prediction 47
5.2 The comparison between spaCy and Flair in NER model training 54
5.2.1 Training cost comparison 54
5.3 Summary 55

6 Conclusion **57**
6.1 Primary findings 57
6.2 Limitation 58
6.3 Contribution to Community and advice for other researchers 58
6.4 Future work 59

List of Figures

2.1 This is example about how does NER work . . . 7
2.2 The components in spaCy processing pipeline . . . 8
2.3 NER example in Flair . . . 9

3.1 A instance of the data we use . . . 12
3.2 Processed data in the second step . . . 13
3.3 Index annotation conceptual model . . . 13
3.4 BIO annotation conceptual model . . . 14

4.1 Annotation method 1 . . . 17
4.2 Annotation method 2 . . . 17
4.3 spaCy training procedure . . . 19
4.4 Diagram for spaCy training process . . . 19
4.5 BIO annotation . . . 20
4.6 Sentences class . . . 20
4.7 A training sample with BIO annotation . . . 21
4.8 Diagram of training proprietary model . . . 23
4.9 Diagram of training joint model . . . 24

5.1 Two kinds of model test on the same test set . . . 30
5.2 5-fold cross validation . . . 31
5.3 Precision result plot of individual models and joint models . . . 34
5.4 Recall result plot of individual models and joint models . . . 34
5.5 F1-score result plot of individual models and joint models . . . 35
5.6 F1-score of proprietary models and joint models in different countries . . . 36
5.7 Probability density plot of the skills frequency distribution . . . 37
5.8 Log probability density plot of the skills frequency distribution . . . 38
5.9 F1-score for different groups in "Certification" prediction in Australia . . . 49
5.10 F1-score for different groups in "Certification" prediction in Canada . . . 50
5.11 F1-score for different groups in "Certification" prediction in UK . . . 51
5.12 F1-score for different groups in "Certification" prediction in Singapore . . . 52
5.13 F1-score for different groups in "Certification" prediction in US . . . 53

List of Tables

4.2 The definition of confusion matrix. 24
4.1 Pros and cons and peculiarities of spaCy and Flair 27

5.1 Data composition 29
5.2 Cleaned data composition 30
5.3 Result of K-fold validation of Australia NER model 32
5.4 Result of K-fold validation of Canada NER model 32
5.5 Result of K-fold validation of UK NER model 32
5.6 Result of K-fold validation of Singapre NER model 32
5.7 Result of K-fold validation of US NER model 33
5.8 Australia model predict on other test sets 36
5.9 High frequency skills in Australia 38
5.10 High frequency skills in Canada 39
5.11 High frequency skills in UK 39
5.12 High frequency skills in Singapore 40
5.13 High frequency skills in US 40
5.14 Missed skills in Australia dataset 41
5.15 Missed skills in Canada dataset 42
5.16 Missed skills in UK dataset 42
5.17 Missed skills in Singapore dataset 43
5.18 Missed skills in US dataset 43
5.19 New skills in Australia dataset 44
5.20 New skills in Canada dataset 44
5.21 New skills in UK dataset 45
5.22 New skills in Singapore dataset 45
5.23 New skills in US dataset 46
5.24 Performance of the **Certification** NER model 47
5.25 Performance of the **Degree** NER model 47
5.26 Performance of the **Email** NER model 48
5.27 spaCy NER model cost and effect 54
5.28 Flair NER model cost and effect 55

Chapter 1

Introduction

We first present the main idea of our work in thesis statement in this chapter. After that, we give a terse introduction about the study in labour market job advertisements, emergent skills detection and extraction. Last but not least, we provide the overall outline for the thesis.

1.1 Thesis Statement

For the job seeker who are concentrating on the searching information from the recruitment advertisements, they will choose online recruitment website as their choice. But how to help them have an overview of labor market, and know which skills are popular ? We build a reliable model to extract the skills emerging in the job advertisements in recruitment website to solve this problem.

1.2 Introduction

The development of computer technology bring change to many industries. As automation increases productivity, shorter work weeks for labour, and reduced factory lead times. It allows more and more functions once performed by humans to be automated. Therefore the jobs of the future will use different skills and may have higher educational requirements. In this episode, it is best for replaced workers to retrained themselves to acquire new skills which allow them to have transitions into new occupations successfully and less likely to be automatized[35]. However, which are the skills in highly demand that one should learn? The online labour platforms, for instance online recruitment websites may hold the key. They match demand and offer of work remotely and it allows an overview of the labour market and longitudinal source to analyse skill dynamics[28]. By reason of the constantly changing employment market, some skills may be out of date and be discarded gradually, while other skills may experience shortage in able workers.

Natural language processing (NLP) technology is actually able to be applied to obtain important skill information from considerable and prolix job advertisements in online labour marketplace. Before we jump into skill information extraction work,

the most essential thing is that the advanced model we designed should have capability that it can understand the content of job ads relatively accurately, so it can process the text in a correct way. Considering the composition of the skills in text, they usually appears in the form of noun phrase like "website management", "customer service", "digital marketing", we spontaneously believe that Name Entity Recognition (NER) is extremely suitable to extract these skills.

Named Entity Recognition (NER) is a subtask of information extraction, it detects the span and the semantic category of entities from a chunk of text[30]. The specific task of common NER can be formalized as a sequence labeling task: a sequence labeling model [8] which is trained to assign a single tagging class to each unit within a sequence of tokens[30] and that is exactly what we mainly focus on and need.

The state-of-the-art of NER evolves from initial stage of dictionary and rule based method to traditional machine learning based method using graphic models including Hidden Markov Model (HMM)[7], Maximum Entropy Markov Model (MEMM)[34] and Conditional Random Field (CRF)[26]. NER benefits from the breakthrough in Deep Neural Networks domain in recent 10 years, realizing a further progress[27]. In this stage of NER development, Recurrent Neural Network (RNN) variants – typically one using a Long Short-Term Memory (LSTM) unit – in combination with CRFs using multiple layers to fetch features in different abstraction levels was introduced and gained quite notable achievement in solving long distance dependencies problems. Further more, the latest method of attention based and semi-supervised learning method may also bring dramatic change in NER. However, the state-of-the-art models are mostly built by supervised learning methods and these models are used in wide-ranging. Therefore, it means the performance of the model is closely associated with the quality and quantity of the training data with labels.[48].

The aforementioned tasks in NLP require reliable training data to produce meaningful results. Therefore we use advertisements from different aggregator websites in different English-speaking countries including UK, USA, Singapore, Canada, and Australia during the time starts from February 2020 to July 2020. After guaranteeing the universality and timeliness of the data, we apply supervised learning method to complete the skill extraction mission.

In the end, based on the performance of our model, we deeply analyze the human labor cost against the performance gain, forecast and estimate the value of our research, and present a conclusion for our research and provide direction in which the researchers are worth spending their time and energy for a further improvement in future.[9]

1.3 Thesis Outline

This thesis consists of 6 chapter in total, these chapters are organized as follows:

- **Chapter 2 Background and Related Work** presents the motivation of this work, introduce the Natural language processing(NLP) techniques Named entity recoginition(NER) and current popular NLP frameworks- spacy and Flair.

- **Chapter 3 Data processing and synthesis** illustrates how to do data processing including collect informative content from XML format raw data, and make annotation for two kind of NLP frameworks.
- **Chapter 4 Method** introduces the method how we deal with the specific problem we met in establish training sets for spaCy and Flair, and the strategy about how to train proprietary model and joint model. We also give the evaluation metric for the model prediction.
- **Chapter 5 Results** talks about the performance of proprietary model and joint model respectively, and the findings during the analysis of these two kings of model. In order to figure out the factors contribute to wrong prediction, we scrutinize the "SKILL" predication result but not restricted on "SKILL" entity prediction we continue to investigate other factors by analyzing other entities, including,"Certification","Degree", "Email".
- **Chapter 6 Conclusion** presents the primary findings in this research, illustrate the limitations of our work and clarify the contribution we made to the research community. Finally we propose the future work for our research.

Chapter 2

Background and Related Work

In this part of the thesis, we will first clarify the motivation behind this research and the meaningfulness as well. Then we will talk about the current method used in Named entity recognition (NER) in order supplementing the background knowledge we used in this research.

2.1 Motivation

It is a fact that we all live in an age with frequent dramatic change. The new technologies and innovative inventions are springing up every year or even every month. With the development of computer technologies, the mode of production in many industries are changing. Automation is reducing the costs of labor and production, at the expense of replacing human labor. Some skills may be gradually dying, accompanying with some skilled workers replaced by the machine.

One possible avenue for replaced workers is to upskill, i.e. to acquire new skills which will allow them to transition into new occupations which are less likely to be automatized. However, what are these emergent skills that one should learn and where can we get the information about these skills are still open questions. The rise of online recruitment websites may provide answer. The online recruitment website can be used as a barometer of the labor market, as it provide information to help people have an overview of the labor market and it is a good source for analyzing skill dynamics. As the employment marketplace is not static, the demand of the skill tends to be volatile: some skills may be outdated, while others may require more able workers. In this research, we will achieve these goals in three steps: 1) Extract the skills appearing in the job abs from the online recruitment websites. 2) Identify the new skills 3) Build a model of which the training time and accuracy are acceptable.

2.2 Related work

We will lanconically present, in this section, some fundamental relevant works in the domain of natural language processing (NLP), techniques of machine learning

prevailed in NLP domain, Named entity recognition (NER) development, and some popular NLP frameworks [11, 24, 25, 36, 37, 46, 58–60, 67, 77–79].

2.2.1 Natural Language Processing

NLP is a subfield of linguistics, computer science, and artificial intelligence, it concentrates on computers and human(natural) language interactions, especially how to enable computers to process and analyze considerable natural language data[75]. Everything we express (either verbally or in written) carries huge amounts of information. As a sender, your task is not only to convey what you want to express, but also ensure that the receiver understands the message you sent. The development of NLP can be divided into three main periods, 1) Symbolic NLP from the 1950s -early 1990s 2) Statistical NLP (1990s-2010s) 3) Neural NLP (present).

There are two main techniques to complete Natural Language Processing tasks - Syntactic analysis and semantic analysis[12]. Syntax means the arrangement of words in sentences or clauses such that they have grammatical meaning in a given language. Syntactic analysis, in NLP, is applied to appraise the underlying rule of association between natural language and grammar rules. Techniques in syntax include lemmatization, part-of-speech tagging, parsing, stemming, etc. In contrast, semantics represents the meaning that the text delivered. Semantic analysis involves applying algorithms in the domain of computer to grasp the meaning and interpretation of words and the rule of sentence structure. The techniques which are widely adopted in semantic analysis include Named entity recognition (NER),Word sense disambiguation,ect. In order to build a comprehensive nlp system, these two parts must be taken into consideration in handling the different language cases.

2.2.2 Named entity recognition

As a secondary task of information extraction, NER aim to identify, locate and categorize named entities contained in text into categories which are defined previously. These catergorizes often includes, locations, date, names, orgnizations, ect[74]. A simple example of NER is showed in Figure 2.1.

NER has always been a research hotspot in the field of NLP. The development of NER starts from early dictionary-based and rule-based methods, to traditional machine learning methods which using Hidden Markov model (HMM), conditional random field (CRF), Maximum-entropy Markov model (MEMM), to deep learning-based methods inluding Recurrent Neural Networks (RNN) + CRF, Convoluted Neural Network (CNN) + CRF. Some recent emerging methods, such as: attention model, migration learning, semi-supervised learning also achieve notable result. [13]

From 2013 to 2014, in this period of time, neural network started to be introduced to solve NLP tasks. Some remarkable achievement like bidirectional Long short-term memory (LSTM) introduced by Graves et al in order to do speech recognition[16], CNN was introduced for modelling sentences by Kalchbrenner[23] changed the mainstream of NLP in a large extend. Benefit from the development of NLP, Deep

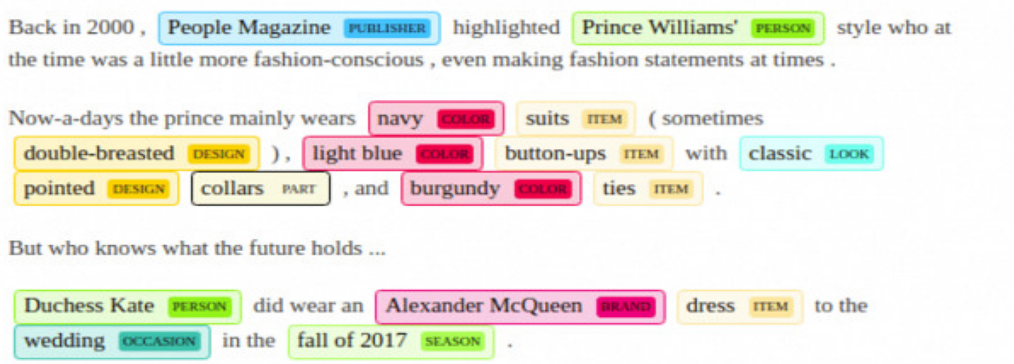

Figure 2.1: This is example about how does NER work

learning-based NER emerged from that time gradually become dominate among the considerable NER model types and in recent years it achieves many state-of-the-art results[29].

Deep learning (DL) is a popular machine learning techinque, that has revolutionised various domains [3, 17, 18, 32, 33, 38, 39, 63–65, 69, 80] The "deep" is from the characteristic that it use multiple layers to process. Data will be learned hierarchically in different abstraction in network[27]. The main advantage of deep learning is its ability of the semantic composition supported by the vector representation as well as neural processing and representation learning[29]. This advantage allows a pattern that one can feed raw data into network and the latent representations and the necessary processing for task such as classification, will be automatically discovered[27]. To explain why Deep learning techniques are applied to NER, there are three core strengths. First, NER can leverage the non-linear transformation in neural network, and a non-linear mappings from input to output is then generated. This characteristic compared with traditional linear model, enables complex and recondite features contained in data to be discovered and learned through non-linear activation functions. Second, deep learning reduce the workload on designing NER features in a large extent. The traditional feature-based approaches requires domain expertise, while the deep learning(DL) models when processing raw data, are effective in automatically mastering data representations. Third, by applying gradient descent, training an end-to-end deep neural NER models become feasible. This property give the developer a chance to design a complex NER model[29].

2.2.3 Popular nlp tools

Because of the fact that Python as a programming language is one of the most apposite for Big Data processing, plenty of tools, libraries and packages are designed for it [22, 32, 39–44, 49, 64, 65, 69]. The NLP library like CoreNLP which is from Stanford group, the Natural Language Toolkit(NLTK), spaCy(an industrial-strength NLP library built for performance), Flair(framework for state-of-the-art NLP) is widely accepted and used among developers who aim to solve problems in NLP domain. We

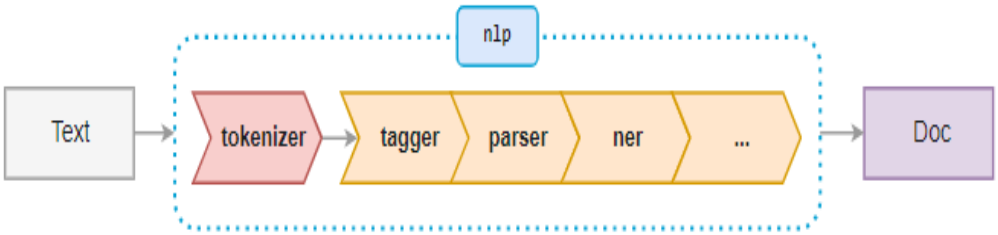

Figure 2.2: The components in spaCy processing pipeline

will highlight spaCy and Flair since they are the NLP frameworks which are adopted in this research project.

SpaCy is an open-sourced library that is free for advanced NLP in Python[20]. It is designed for production use and build applications that is able to process and "understand" large volumes of text. SpaCy has been widely applied to build the systems used for information extraction or natural language understanding, also it is popular for doing text pre-processing in deep learning tasks. The spaCy includes some features and capabilities like Tokenization, Part-of-speech (POS) Tagging, Dependency parsing, Lemmatization, Named Entity Recognition (NER), Rule-based Matching, Text Classification, etc. Among them, some are concepts in linguistic domain, others are associated with the functionality in general machine learning. These features, for most of them, work independently, while some have the requirement that you need to load statistical model. These models allow spaCy to predict the part of speech – for example, distinguish verbs or nouns. When the model is loaded, it will return a language object called `nlp`. If `nlp` is called to be applied to text, a `Doc` object is produced after spaCy firstly tokenizes the whole text. The `Doc` is processed in different steps subsequently. These processing steps are summarized and also referred to as the processing pipeline(Fig 2.2). In default model of spaCy, its processing pipeline includes a tagger, a parser and an NER component. For every component in the pipeline, after its processing task, it produces a processed `Doc`, and that `Doc` will be continued passed on to the subsequent component.

SpaCy has a simple classifier for its NER model. For instance, a shallow feedforward neural network with a single hidden layer[62]. Even though it has only one hidden layer in the classifier, before any input features are fed into the classifier, spaCy uses a stack of weighted bloom embedding layers merge neighbouring features together. The embedding will form a unique representation to each word for the different context it is in. Therefore, the classifier is made powerful using this clever feature engineering[20].

Flair is another powerful NLP library which is open-sourced and developed by Zalando Research[5]. The central idea of Flair is providing a simple interface for conceptually different embeddings for words and documents[4]. The framework of Flair is built directly on PyTorch which is known as a great deep-learning library[50].The

framework also has the function of standard model training also is equipped with pre-trained model allowing researchers to use state-of-the-art NLP models and make some adjustments according to their demand in their applications[4]. It can afford plenty of NLP tasks including NER. Figure 2.3 is used to illustrate how flair process a sentence.

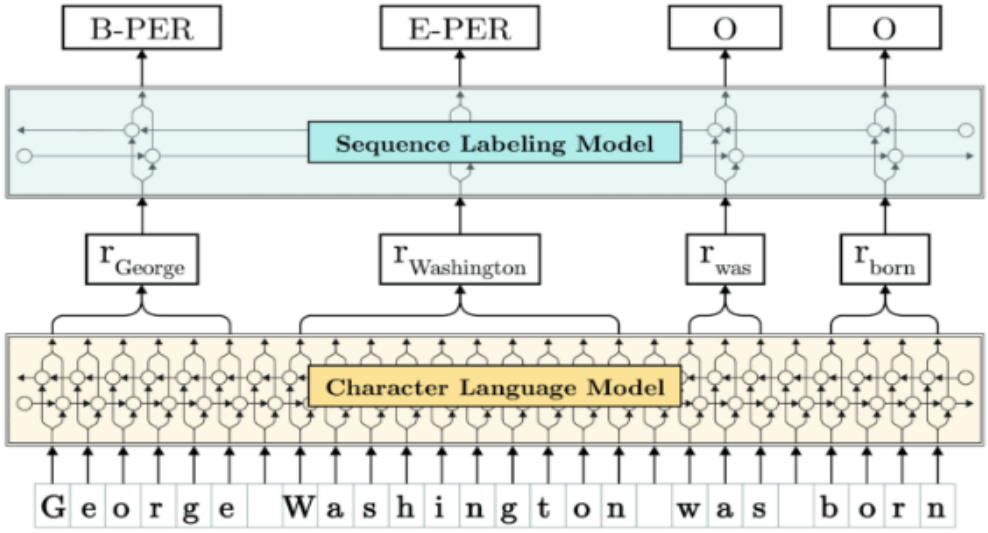

Figure 2.3: NER example in Flair

There is a sentence at the bottom of this diagram. As a character sequence, it is input into a pre-trained bidirectional character language model (LM) whose color is yellow in the figure. This pre-trained model is trained on enormous text corpora without any labels. Flair retrieves a contextual embedding for each word by extracting the first and last character cell states from the language model. Afterwards, the word embedding will be put into a vanilla BiLSM-CRF sequence labeller whose color is blue, in order to generate solid state-of-the-art results on downstream tasks like named entity recognition[4].

2.3 Summary

Both spaCy and Flair are outstanding NLP frameworks implemented in Python despite having different fundamentals. These differences will lead to different training requirement, costs and performance which are suitable to be aligned to compare. In this research, we will apply spaCy to first since it is known as the fastest NLP framework which preserving accurate syntactic analysis[20]. After successfully implement spaCy, we will try Flair and make comparison with spaCy.

Chapter 3

Data processing and synthesis

In this chapter, the data we used in this research will be introduced and we give illustration about how the data can be processed into the format we desired which is able to be input into spaCy and Flair framework. Finally, we will show one instance in the processed dataset.

Recall the goal of this research, we aim to identify the emerging skills in the labour market. Online recruitment websites that can give us adequate job advertisements containing different skills are more than suitable to be our data source. Our data is provided by a researchers from University of Technology Sydney (UTS). These data are from various sorts of websites in five countries that take English as their native language, including Australia, Canada, United Kingdom, Singapore, United States. These websites are reliable, frequently updated and have good reputation in the online recruitment industry, for instance, SEEK, the Australia number one online employment marketplace concentrating on increasing the possibility that jobseekers get their ideal job and helping hirers to find qualified people for advertised role.[1]

The instance of data includes the content of the job ads as well as some XML labels containing part of the job ads text. The sample data is shown in Fig 3.1

3.1 Data Process

In the raw data, text fragments of job ads are categorized in different types and saved in different pre-defined XML format labels. The key information that we are looking for is preserved either by the text inside the particular XML label or the XML label attributes such as `<cannonskill name='English'>`. What we desire to do is that we process the raw data in order to make it can be accepted by spaCy and Flair frameworks. We will make our training set using 3 steps:

- Extract useful data information from the raw data. We use XML ElementTree to represent the whole XML text as a tree, since XML is an inherently hierarchical data format.[2]The redundant nodes are filtered while we store informative texts and attributes in `<canonskill>`, `<skill>`, `<skillcluster>`. After that, we can combine these informative texts which are actually skills appearing in the job ads with the corresponding contents of job ads.

Job ads Content

Clean Pursuit are currently looking for experienced cleaners to join our growing team, To fulfil the role you will need Drivers License Own Car Own Cleaning Equipment (Vacuum/Mop/Etc) Police Clearance ABN At least 1-2 years cleaning experience essential Great communication skills Great time management skills Fluent in English We are looking for experienced cleaners that want a long term, stable position within the company. We are searching for those that have a friendly and accomodating attitude and have flexible work hours. We are a growing company and you will enjoy a great work culture, great rates, regular shifts and the flexibility of the job. We serve - Metro Adelaide (Anywhere within 30km's of CBD) Job Types: Full-time, Part-time, Subcontract Experience: housekeeping: 1 year (Required) License: Australian National Police Check (Required) Australian driver's license (Required) Work Eligibility: The candidate can work permanently with no restriction on hours (Preferred)

parsed result

{"status":true,"statusCode":"OK","requestId":"acf2b79d-5f80-444b-975f-fd231c2e9df8","timeStamp":"2020-07-02 20:03:03.612","responseData":"<?xml version=\"1.0\" encoding=\"UTF-8\" standalone=\"yes\"?>\n <JobDoc>\n <posting blacklist=\"false\" canon-cardinality=\"1\" canonversion=\"2\" cardinality=\"singular\" dateversion=\"2\" experience-canonlevel=\"1-6\" experience-level=\"mid\" experience-max=\"24\" experience-min=\"12\" green-industry=\"false\" iso8601=\"2020-07-02\" minyrsexp=\"2\" present=\"737610\">\n <qualifications>\n <required>\n <certification>you will need Drivers License Own Car</certification>\n </required>\n <required>\n <interval>\n <skill>time management</skill>\n <duration>At least 1-2 years</duration> cleaning experience essential Great communication skills Great skills Fluent in English We are looking for experienced cleaners that want a long term, stable position within the company</interval>Police Clearance ABN .</required>To fulfil the role Own Cleaning Equipment (Vacuum/Mop/Etc) </qualifications>\n <qualifications>\n <required>\n <certification>(Required) Licence</certification>\n <interval>\n <duration>1 year</duration>\n </interval>Subcontract Experience: housekeeping: </required>\n <required>Australian National Police Check</required>\n <required>Australian driver's licence (Required) Work Eligibility: The candidate can work permanently with no restriction on hours (</required>\n <preferred>Preferred)</preferred>: (Required) \"</qualifications>\n <duties>\" Clean Pursuit are currently looking for experienced cleaners to join our growing team,</duties>\n <duties>\n <jobtype hours=\"fulltime\">Full-time</jobtype>\n <jobtype hours=\"parttime\">Part-time</jobtype>Job Types: , ,</duties>\n <background>We are searching for those that have a friendly and accomodating attitude, and have flexible work hours. We are a growing company and you will enjoy a great work culture, great rates, regular shifts and the flexibility of the job. We serve - Metro Adelaide (Anywhere within 30km's of CBD)</background>\n </posting>\n <special xsi:type=\"jsonObject\" xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\"/>\n <skillrollup version=\"1\">\n <canonskill name=\"Communication Skills\" skill-cluster=\"Specialised Skills\">\n <variant>Great communication skills</variant>\n </canonskill>\n <canonskill name=\"English\" skill-cluster=\"Specialised Skills\">\n <variant>English</variant>\n <variant>English</variant>\n </canonskill>\n <canonskill name=\"Equipment Cleaning\" skill-cluster=\"Maintenance, Repair, and Installation: Equipment Repair and Maintenance;Specialised Skills\">\n <variant>Cleaning Equipment</variant>\n </canonskill>\n <canonskill name=\"Time Management\" skill-cluster=\"Specialised Skills\">\n <variant>time management</variant>\n </canonskill>\n </skillrollup>\n <DataElementsRollup version=\"5.15.0.22 Optic v3.4.3.9\">\n <YearsOfExperience>At least 1-2 years|1 year</YearsOfExperience>\n <CanonYearsOfExperience>\n <min>12</min>\n <max>24</max>\n <level>mid</level>\n <canonlevel>1-6</canonlevel>\n </CanonYearsOfExperience>\n <Skills>Great communication skills|English|English|Cleaning Equipment|time management</Skills>\n <CanonSkills>\n <canonskill name=\"Communication Skills\" skill-

Figure 3.1: A instance of the data we use

- Delete duplicate job ads. Because there are some companies post their recruitment advertisements in many different website, we delete duplicate job ads and only keep the latest one. Then we assign every job ads descriptions and the skills an unique ID to identify them. We use DataFrame to store the data which is clear and convenient to search particular content.

JobID	Skills	Job_descriptions
2af4dbcb1a3f46fb	[cardiopulmonary resuscitation (cpr), planning...	short description this exciting role assists ...
6697f622ba197a5c	[cooking, food safety, hazard analysis critica...	closing date 12-jul-2020 employment type full...
c6d4da647d94cf27	[care planning, communication skills, data ana...	qld state office | oversee a portfolio of bri...
6bb6357f674af943	[communication skills, costing, detail-orienta...	this role reports to the national supply chai...
0ad3a73f95f9c154	[detail-orientated, guest services, microsoft ...	job title guest service agent job description...

Figure 3.2: Processed data in the second step

- Make entities annotation. We plan to use spaCy and Flair framework to train robust NER models respectively. However, the enetity annotation methods for the training set of these two frameworks are quite different. SpaCy accept index annotation in input training set while Flair use annotation with IOB tag. For index annotation, the position indices of the entities should be annotated along its label.

```
train_data = [
    ("Uber blew through $1 million a week", [(0, 4, 'ORG')]),
    ("Android Pay expands to Canada", [(0, 11, 'PRODUCT'), (23, 30, 'GPE')]),
    ("Spotify steps up Asia expansion", [(0, 8, "ORG"), (17, 21, "LOC")]),
    ("Google Maps launches location sharing", [(0, 11, "PRODUCT")]),
    ("Google rebrands its business apps", [(0, 6, "ORG")]),
    ("look what i found on google! 😂", [(21, 27, "PRODUCT")])]
```

Figure 3.3: Index annotation conceptual model

IOB tagging is a tagging format which is used to tag tokens. Similar to part-of-speech tags, IOB tagging provides tags indicating inside, outside, and the beginning of a chunk.[55] The example of IOB scheme is like this. The B-prefix associated with a tag represents that the is word is the beginning of a chunk(here the chunk is the skills we extracted from the job ads), and an I-prefix associated with a tag represents that the tag is inside a chunk if the length of the chunk is greater than one. Other unrelated tokens will be annotated with

O means they belongs to no chunks.[56][71]

```
Alex I-PER
is O
going O
to O
Los I-LOC
Angeles I-LOC
in O
California I-LOC
```

Figure 3.4: BIO annotation conceptual model

3.2 Summary

In this chapter, we display the raw data and introduce which steps we should follow in order to transform the data into our desired format generally. The essential steps can be summarized as skill extraction, text concatenation(skills and advertisement content) and duplicate removal, entities annotation. In the next chapter, more details in research methods and minor adjustments in terms of these processing steps will be revealed.

Chapter 4

Method

We will illustrate the methods we used in different phases of the whole research in this chapter, from data acquiring and processing to model training and to result evaluation.

4.1 Data acquiring and evaluation

As what we discussed, considering our final goal is to enable a NER model to identify the skills(The skills are already exist or emerging in the real world) precisely and quickly, we need job advertisements working as our training set so that strengthen the model ability to find skills. We use the data from a researcher in UTS and the format of the instance of data is already shown in Fig 3.1

It is easy to find the valuable information is wrapped by XML lables named `Canonskill` or `skill`, etc. We utilize the Python built in library, `ElementTree`, to parse XML data. After initializing a tree structure for a XML data, we can recursively search the text and attributes of the children nodes which contains informative skill message. Combine the skills with corresponding job ads text along with its unique jobID, we obtain the preliminary data. The preliminary data is actually problematic due to some flaws - duplicate data, different spelling habit, some non-English content, upper case and lower case. Duplication and non-English content is easy to solve by just deleting the data from the whole preliminary dataset since they provide noise in training NER model.From our observation, skills extract from the XML label tends to have two kinds, use skill **customer service** as an example, one kind would be **Customer service** and the other is **customer service** due to their position in the text. Because we will annotate the skills in subsequent step, and we expect the model improve the ability to find skill regardless it form, we transform all the job ads content into lower case, therefore the case will not cause problem in annotation and training. Our data from five different English speaking countries leading to the British and American spelling problem. Our solution is that rather than force two kinds of spelling into one kind since it is quite time-consuming to find all the words with spelling problems, we training two types NER model, one is a joint model which use training data consist of five countries. Apparently, two kinds of spelling format are both included in the training data. The other one is the proprietary NER model for

each countries which only use the job ads from one specific countries to train, such as only use the training data from Australia to train the Australian proprietary NER model, through this we ensure the NER model trained by one specific countries get rid of the influence from other spelling format. Why we propose two kinds of NER model? Recall our goal, We want to build a reliable model to extract emerging skills from job ads whether the advertisements use British spelling or American spelling, it accentuates the covering range. While, if we have a clear objective, for instance, we just want to have a overview on the emerging skills in online recruitment website in UK, the proprietary model for UK will generally perform better than the joint model. This will be formally introduced in result part along with our experiment result. After solving the intrinsic problems in the preliminary data, we can use these data to create our training sets for two Framework, spaCy and Flair. These two frameworks both allow us to train custom NER model and that characteristic meets our requirement of find "Skill" entity from different text.

4.2 Framework comparison and selection

Although our task is transforming the cleaned data into two different training sets that can be accepted by two frameworks, the difference of these two kinds of training set only exists in the entity annotation methods. We will discuss the procedure of create two training sets and the peculiarities of them respectively.

4.2.1 Creating spaCy training set and spaCy model training

As we mentioned before, spaCy requires index entity annotation when training NER model. We already have large amount of preliminary data, each of them consists of job ads text and their corresponding skills contained in the text which we denoted as entities. What we need to do is to find the start indices and end indices for each of these entities in the text and give these pair of indices a "SKILL" label.

However, there is a intrinsic limitation of training custom NER model using spaCy. The named entity recognizer is constrained to predict only non-overlapping, non-nested spans in spaCy, therefore, the training data should obey the same constraint[20]. If we check the skills list in the cleaned data some nested skills such as "Social Media" and "Social Media Planning", "Digital marketing" and "marketing" can be found. In order to solve this problem, we propose two kind of solution at first. One is we preserve the longer one that is to say preserve "Digital marketing" and discard "Digital"(fig 4.1), another option is we could have two same text but with the different annotations(fig 4.2).

It is not hard to discover that when we have two entities overlapped, the longer one tends to be more detailed and include more information compared with shorter one. In the example of "Digital marketing" and "marketing", the former is specific underscoring that it is "Digital" yet the later is more general and wide-ranging in terms of describing the skill. Considering we aim to capture more informative skills

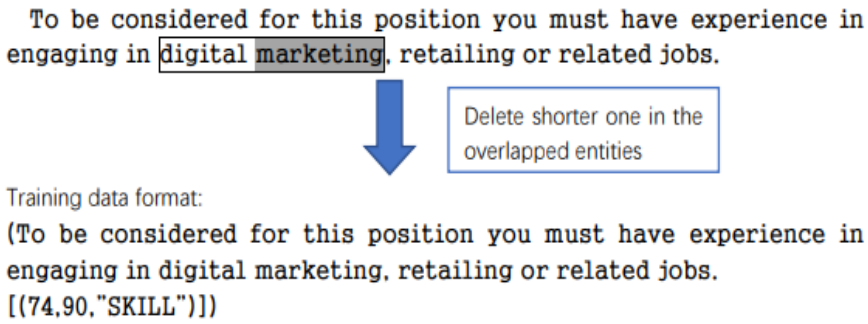

Figure 4.1: Annotation method 1

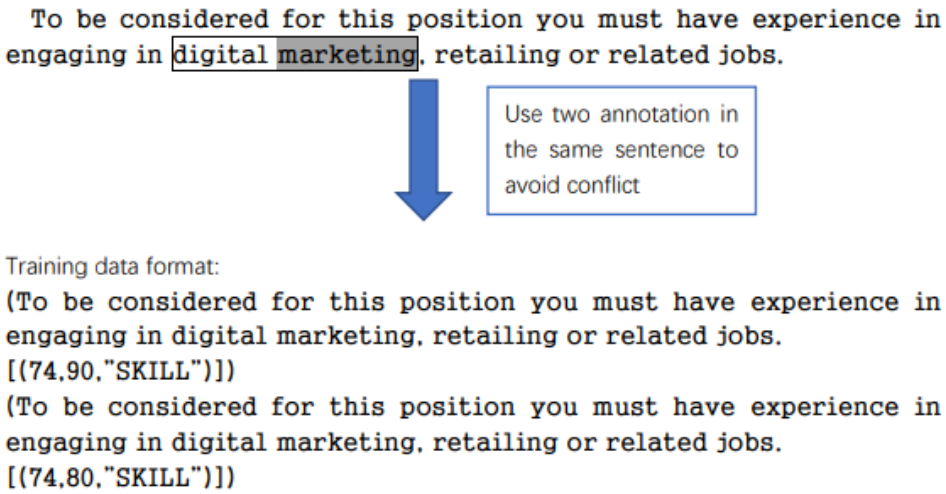

Figure 4.2: Annotation method 2

in a certain job ad, we can simply delete the shorter one. The advantages of the this method is that it solves the overlapped problem by deleting the shorter entity when overlapped happens and does not increase the workload in training process at the meanwhile. Yet it indeed exacts a toll on losing some information to some extent due to deleting the shorter entity.

The second choice is that we give up deleting any of the overlapped entity but annotate them in two identical text respectively. By applying this method, we need not to worry out the information loss since do not delete anything. Nevertheless, repetitious text for rare pairs or even just one pair of overlapped entities cause redundancy in training also it is unknown whether it will help or hurt the performance of your model depend on different tasks[19].

Although the second method does not lose information, copying the whole text for small number of overlapped entities seems unnecessary, especially considering the uncertainty it may cause to the performance of prediction and the manually increased training time. According to our observation, the amount of the more general

one entity in a overlapped entity pair is small therefore it only cause trifling information loss if we delete them. In the light of above, we finally decide to adopt the first method to deal with the overlapped problem.

So far, we solve the problem brought by nested entities, which means the dataset is now able to be accepted by spaCy. Next we will introduce the more details in training a NER model.

The named entity recognizer model trained by spaCy is a statistical model, the output about which part should be assigned a label or whether the word is a named entity in the prediction. Like other supervised machine learning techniques, you need to "tell" spaCy the example which is a pair consisting of an input object (In our case, the text in job ad) and a desired output value (The words which are "skill" named entities existing in the text)[21].Then the spaCy will make prediction based on the training data. When exposed to unlabelled data, the model will make prediction based on what it mastered during the training. As the matter of fact, we know what the correct answer is, thus we can give feedback to the model about its prediction in the form of the loss function[20]. The loss function in statistics is used for parameter estimation, and measure the difference between true and estimated or predicted values for data instance[72].The loss function that spaCy adopted is log loss function[73]The spaCy desire to do optimization by minimizing the loss function which means the predicted value is close to the true value for the instance of data.

$$L(Y, P(Y|X)) = -logP(Y|X) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij} log(p_{ij}) \tag{4.1}$$

The loss function will be simplified if we do binary classification. In our research, during the training, the model will predict whether this word in the text belongs to our predefined entity which is binary classification task.

$$L = -\frac{1}{N}\sum_{i=1}^{N}(y_i log(p_i) + (1 - y_i) log(1 - p_i)) \tag{4.2}$$

In the simplified log loss, y_i is the correct value associated with input,P_i is the probability of the prediction is desired named entity generated by the model[51].

The spaCy will optimize its model through iterations specified by hyperparameters, the greater the difference, the more significant the gradient and the updates to our model. Here we use a diagram of the process flow to make the whole training process more intuitive.

In training, SpaCy has a object called `GoldParse` to collect the annotated training example. This object is initialized with the `Doc` object it refers to, some keyword arguments such as **tags** or **entities** are used to specify the annotations in the text. Be more specific, the job of `GoldParse` is about annotations encoding, and creating data structures which is needed for access. With the `Doc` and its gold-standard annotations, the model can be updated wisely and be capable of learning sentence of many words with their labeled named entities.

When training a NER model, what we are most reluctant to see is that the model to

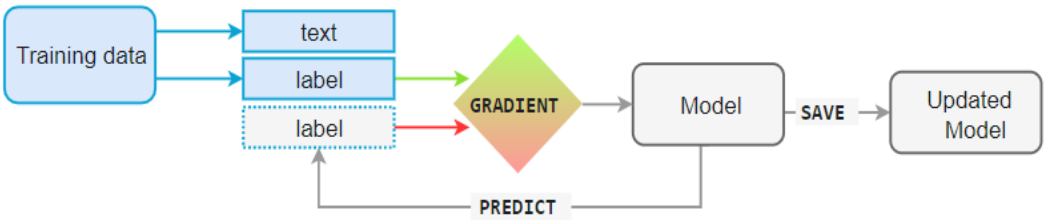

Figure 4.3: spaCy training procedure

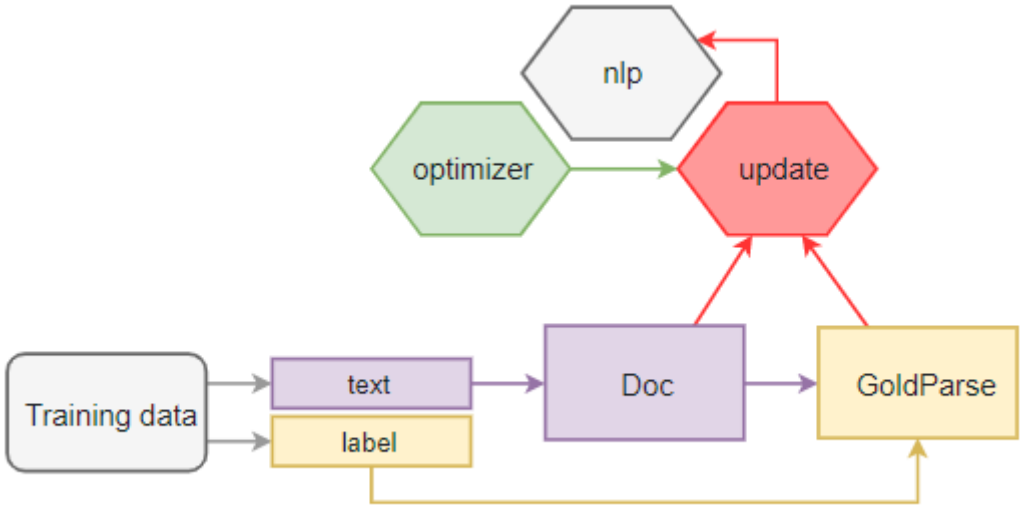

Figure 4.4: Diagram for spaCy training process

memorize the examples we offered, we expect it to "summarize" a series of theories that can be generalized among other examples that it has never met. Specifically, if model to learn that in this instance of “Washington”, it is a location, we want it to learn that “Washington”, is most likely a location rather than a name of a person in contexts like this[57]. Because of this, the model training process put forward higher requirement of the training set. NER has become more of a data problem than an algorithm problem[14]. That’s the reason that we tend to select the representative training data which can reflect we want to process[20]. During the training, only show a model a single example once is far from enough. The solution is that we train iteratively to make sure the model can "learn" as much as possible. The training data is shuffled in each iteration, then the model doesn’t make generalizations under the influence of the order of examples[47]. There is another trick to improve the learning results, setting a dropout rate, which randomly “drops” individual features and representations during the training. Dropout rate prohibits the model from memorizing the training data which is what we devote to avoid. If we set 0.25 as our dropout

rate, it means that data in the training set has a 1/4 likelihood of being dropped[68].

After taking these technique which can possibly improve our model performance into consideration, we can now put our hands to training.

4.2.2 Creating Flair training set and Flair model training

The entity annotation in Flair should follow the format like Fig 4.7.

```
George N B-PER
Washington N I-PER
went V O
to P O
Washington N B-LOC

Sam N B-PER
Houston N I-PER
stayed V O
home N O
```

Figure 4.5: BIO annotation

With code readability and ease-of-use in mind, Flair represents NLP concepts such as tokens, sentences and corpora with simple base (non-tensor) classes called **Sentence**[4].

```
# init sentence
sentence = Sentence('I love Berlin')
```

Figure 4.6: Sentences class
[4]

Therefore, We can regard the whole cleaned data as a corpus which includes many Sentences. We will use an empty line to separate different sentences in the corpus and each line has three columns. The first column represents word, the second column is the corresponding POS tag and the final column denotes the BIO-annotated

NER tag(in our research, this tag is "SKILL" with a BIO prelix). In the Flair annotation part, we encountered the same problem - nested entities as we ever met in spaCy annotation, we applied the same strategy which preserving the longer one and delete the shorter. We still use nested entities "marketing " and "digital market" to demonstrate Flair annotation. From Fig 4.9 you can find we annotated "marketing" as a word inside a chunk rather than a word the beginning of a chunk.

```
experience N O
in P O
engaging V O
in P O
digital A B-SKILL
marketing V I-SKILL
```

Figure 4.7: A training sample with BIO annotation

The core concept of Flair is Embedding, a NLP technique to map words or phrases from the vocabulary to vectors of real numbers[10] which captures both semantic and syntactic information of words[31]. Pre-trained classical word embeddings have been shown to bolster downstream NLP tasks, because of their ability to improve learning result and generalization with information learned from data, and the easy deployment in pervasive learning approach.[45]. In this research we will choose a classic word embeddings called **GloVe embedding** to map the word.

In recent 5 years, the GloVe word embedding become the most outstanding and popular embedding.It is a global log-bilinear regression model, combining the advantages of global matrix factorization and local context window methods these two major model families[52]. The reason we decide to use GloVe embedding not only is its ability to capture the useful linear substructures that is popular in early methods like Word2Vec[45] but also our corpus incorporate considerable words making our machine hard to process when using other kinds of embeddings.

4.2.3 Summary of spaCy and Flair

Preparing training set for spaCy and Flair, as a matter of fact, is two independent tasks since spaCy and Flair are different in their inner structures, as well as their input requirements. To be more intelligible for the people who did not familiar with these two frameworks, we made a comparison between spaCy and Flair, summarized and listed their advantages and disadvantages and charateristic in Table 4.1

Our main tool for this research is spaCy as we portrayed it is a reliable NLP framework. Flair is not mature at present compared with spaCy, the training time of Flair is very long and it demands powerful hardware to support training. Owing to the lim-

itation of time and device, we choose to train considerable NER models using spaCy not Flair. Therefore you will see large number of result from the models trained by spaCy but relatively small number of result from the model produced by Flair in the subsequent **result** part.

4.3 Model training

At present, we clarify the responsibility of this two framework that spaCy is the mainstay and it plays the necessary role in this research and yet Flair only shoulder relatively small responsibility in this research. Based on the data we have, we propose two kinds of training strategy, as we explained previously, one is that training proprietary model for each country, the other is we train a joint model for five countries.

4.3.1 Proprietary model

Our training data consist of job advertisements from five English speaking countries, Australia, United Kingdom, Canada, Singapore, United States. The data in each country can be split into training set and test set in the ratio of 7:3 after randomly sorted. Then five training sets containing job advertisements from five countries will be used to train five **proprietary** NER models respectively as the Fig 4.10 shows.

So far, we have 5 proprietary models and these models will be expected to make prediction on their corresponding test sets.

4.3.2 Joint model

The second type of model called **joint model** is trained by a joint training set which is a combination of five training sets and the data from five countries is randomly split at ratio of 70%. Instead of directly combining all the job advertisements from five countries together and then split into training and test set at ratio of 7:3, this method adopt first split then combined strategy because the data from each country is not uniformed and exist difference in quantity. If we first combine them together and then split, it may cause the problem that the data from a certain country accounts for large proportion in the training set after split due to its larger amount of data compared with other countries. The residual test sets from each country will be combined into one joint test which will be used for subsequent model test. The conceptual model of training joint NER model is presented in Fig 4.11.

The prediction result from these two different kinds of model gives us information and we leverage them to analyze advantages and disadvantages of two kinds of model and summarise their peculiarities. In **result** part, we will reveal more details and make explicit analysis.

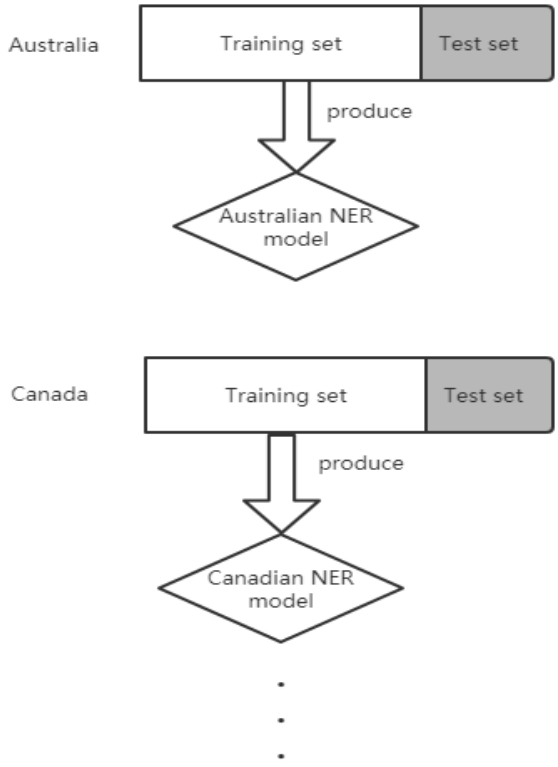

Figure 4.8: Diagram of training proprietary model

4.4 Evaluation methods

In this section, we will introduce the evaluation method **confusion matrix** which can reflect the performance of our models in a straightforward way.

Confusion matrix, a specific table layout that offers visualization of the performance of algorithms which aims to solve classification problems in machine learning domain particularly in the supervised learning[66][70].The actual and predicted classifications done by a classification model is recorded in the confusion matrix[61].The columns in the matrix are used to represent the instances in the actual class, while the row represents an instance predicted by the model (predicted class), or vice versa[53]. As the name "confusion matrix" suggests, the function of matrix is to check whether the prediction system is confusing two classes or in another word, commonly mislabeling one as another.

Confusion matrix is a special instance of contingency table, with "actual" and "predicted" two dimensions, and in both dimensions there are identical sets of "classes" , the dimension and the class form totally four kinds of combinations and each of them is the variable in the contingency table [70].

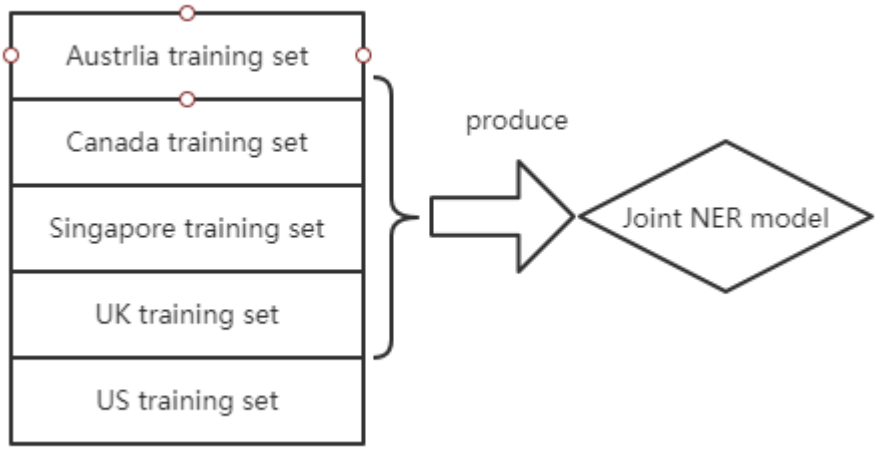

Figure 4.9: Diagram of training joint model

		Actual class	
		Actually Positive	Actually Negative
Predicted class	Predicted Positive	True Positive (TP)	False Positive (FP)
	Predicted Negative	False Negative (FN)	True Negative (TN)

Table 4.2: The definition of confusion matrix.

In table 4.2, the column labels "Actually Positive" and "Actually Negative" refer to the ground-truth labels in your data set and the row labels "Predicted Positive" and "Predicted Negative" refer to your model's predictions, i.e. what your model thinks the label is.

Note that the entries inside of a confusion matrix (TP, TN, FP, FN) are counts[54]:

- **True Positives (TP)**: The number of instances that the model predict as positive and the prediction is correct(True).
- **True Negatives (TN)**: The number of instances that the model predict as negative and the prediction is correct(True).
- **False Positives (FP)**: The number of negative instances that the model incorrectly classified as positive (i.e. the negative examples that were falsely classified as "positive")
- **False Negatives (FN)**: The number of positive instances that the model incorrectly classified as negative (i.e. the positive examples that were falsely classified as "negative")

Now can use these four entries to define some helpful measurement.

1. **Precision**: In classification tasks, precision for a class shows the ratio between the number of instances that are correctly classified as positive(true positive) and the total number of instances that are predicted as positive by the model(the sum of true positive and false positive).

$$Precision = \frac{TP}{TP + FP} \tag{4.3}$$

2. **Recall**: Recall in this context is defined as the number of instances which are correctly classified as positive(true positive) divided by the total number of elements that actually belong to the positive class (the sum of true positives and the number of the instances that are misclassified as negative but in fact postitive i.e. false negative).

$$Recall = \frac{TP}{TP + FN} \tag{4.4}$$

   To be more specific, we give a concrete example. In a classification task, if the precision score for a class A means is 1.0, it represents that each item labelled by the model as belonging to class A does indeed belong to class A, while it indicates nothing about the number of instances in class A that were labelled incorrectly. If the recall of this class A equals to 1.0 means that all instances from class A was predicted by our model as belonging to class A however, it provides no information about the amounts of the instances from other classes were falsely also labelled as belonging to class A[15][76].

   Generally, in evaluation part, precision and recall scores are frequently discussed together. It is often the case that one measure is fixed and compare the value of the other one (e.g. compare recall and fix the precision level at 0.7) or combine precision and recall into a new measure. Example of measures that are a combination of precision and recall are the F-1 score, a harmonic mean of precision and recall.

3. **F1**: In binary classification task in the domain of statistical analysis, the F1-score or known as F-measure is calculated from the precision and recall of the test which is used a measure of a test's accuracy

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN} \tag{4.5}$$

   The highest possible value that can be reached of an F1-score is 1, reflecting the classification obtain perfect precision as well as recall, and the lowest possible value of F1-score is 0, when any one in precision and recall is zero.

4.5 Summary of this chapter

In this chapter, we divide the whole process of the research into different stages. We devote to provide solution and rationalize our choice for the problem we met in these three stages. We justified why we preserve the longer entities and delete the shorter one when we have nested entities in the annotation part, we compare the divergence between spaCy and Flair framework in terms of usability, efficiency and performance in order to select one of them to be our main tool, then two kinds of training methods were introduced. Finally, we adopt confusion matrix and elicit several measurements to evaluate our model. In the next chapter, the result of the research will be presented and analysed.

spaCy		
Advantages	Disadvantages	characteristic
1.Well documented and engineered and widely recognized by users 2.Known as the fasted NLP framework regarding training and making prediction 3.Easy to learn and use due to its feature of one single highly optimised tool for each task	1.Accuracy is not significant and limited	1.A shallow feedforward neural network with a single hidden layer 2.Indices annotation: Indices annotation
Flair		
Advantages	Disadvantages	characteristic
1.Open source library designed to reach the state of the art in NER 2.Incorporate state-of-the-art word embeddings	1.Known to be slow 2.Not completed as spaCy, still need a lot of work to prove	1.Deep-learning framework with BiLSTM-CRF sequence labeller 2.Entity annotation type: BIO scheme

Table 4.1: Pros and cons and peculiarities of spaCy and Flair

Chapter 5

Results

Please be advised that two different model training methods were only applied to spaCy framework since the workload of annotation for Flair is heavy and the cost of training custom NER model using Flair is quite high, thus only use two third of our entire data were used to train a joint model using Flair.

In this chapter, the research result will be summarized and organized into the following aspects:

1. The performance of proprietary model and joint model trained by spaCy
2. The interesting findings we obtained from the prediction results
3. The performance of the NER models which used to extract other entities and possible factors leading to wrong prediction
4. Comparison between Flair joint model and spaCy joint model

The development environment and testing environment of spaCy NER model is my own machine which possesses intel i9 9900K as CPU, Nvidia RTX 2070 super as graphic card with 8G memory, 32 GB RAM. The development environment and testing environment of Flair NER model is on the high performance computer(ihpc) in University of technology Sydney with 384 GB RAM and Tesla V100 as graphic card. The composition of the data (job advertisements) we used from online recruitment website is listed below in table 5.1. These are raw data which means the non-English

Country	Amount
Australia	7391
Canada	8175
Singapore	6778
UK	9057
US	5595
Total	36996

Table 5.1: Data composition

job ads, duplicate job ads are included. After our processing and annotating, the amount of data for each country is presented in table 5.2

Country	Amount
Australia	7131
Canada	7766
Singapore	6667
UK	9004
US	5467
Total	36035

Table 5.2: Cleaned data composition

Cleaned data gets rid of the impact of the problems of nested entities, duplicate text and is able to be directly made into training data with proper annotation. From these two tables, only small number of data exists the above problems which does not cause significant information loss and ensure sufficient training data.

In **Model training** part, we mentioned that when making proprietary NER models, we create test sets(proprietary test set) for each country at the same time. In order to compare the performance of the joint model and proprietary model, we tested them on the proprietary test set respectively as the Fig 5.1 shows.

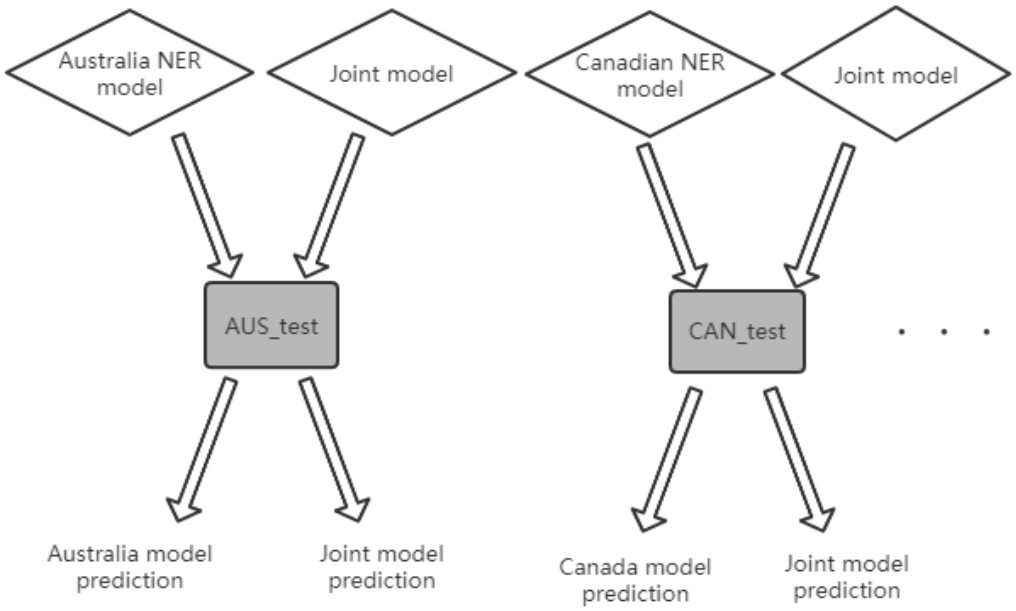

Figure 5.1: Two kinds of model test on the same test set

In the following content, we will firstly show the validation of our proprietary NER model and then compare the predication performance of proprietary NER model

and joint model.

5.1 Model performance

5.1.1 Validation of our training strategy

To prove the validation of our method to train NER model and the reliability of our model(whether proprietary or joint), we applied cross validation to make assessment of the models. Cross validation also known as **rotation estimation** or **out-of-sample testing**, is a statistical technique for assessing how the results of a statistical analysis will generalize to an independent data set[6]. In the goal of the task in predication, researcher would like to evaluate how accurately a predictive model will perform in practice, and cross validation is frequently used. K-fold cross validation is what we adopted in this research which is a special case for cross validation. In K-fold cross validation, this procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into, and k also indicate the times of iteration over a dataset set k times. In each round, we split the whole dataset into k groups: one group is designed for validation, and the residual $k-1$ groups are merged into a training set. The figure 5.2 below demonstrates the process of 5-fold cross-validation:

Figure 5.2: 5-fold cross validation

We use precision,recall and F1 score as the indicator of the performance of models in 5-fold cross validation. For simplicity, we use country abbreviation plus iteration number to represent the model and the performance result for models of each country are present in the tables below.

Model	Precision	Recall	F1 score
AUS_1	93.4	91.4	92.4
AUS_2	92.2	92.4	92.3
AUS_3	92.5	92.2	92.3
AUS_4	93.0	91.8	92.4
AUS_5	92.8	93.3	93.0

Table 5.3: Result of K-fold validation of Australia NER model

Model	Precision	Recall	F1 score
CAN_1	94.5	94.3	94.4
CAN_2	93.4	95.1	94.2
CAN_3	92.2	95.4	93.8
CAN_4	94.8	94.2	94.5
CAN_5	93.8	94.6	94.2

Table 5.4: Result of K-fold validation of Canada NER model

Model	Precision	Recall	F1 score
UK_1	93.0	94.6	93.8
UK_2	94.4	93.7	94.1
UK_3	91.3	94.6	92.9
UK_4	94.3	93.4	93.8
UK_5	94.1	93.6	93.8

Table 5.5: Result of K-fold validation of UK NER model

Model	Precision	Recall	F1 score
SG_1	94.0	94.9	94.4
SG_2	94.9	94.3	94.6
SG_3	92.5	94.9	93.7
SG_4	94.6	94.6	94.6
SG_5	93.6	94.6	94.1

Table 5.6: Result of K-fold validation of Singapre NER model

Model	Precision	Recall	F1 score
US_1	92.3	92.1	92.2
US_2	93.7	92.2	92.9
US_3	93.6	91.6	92.6
US_4	93.0	91.0	91.9
US_5	91.4	92.4	91.9

Table 5.7: Result of K-fold validation of US NER model

If we focus on the F1-score of in these tables, we are able to find that the models in Singapore and Canada achieve a higher F1-score at round 94 percent in predication compared with other countries. Even though minor difference in F1-score exists among these models, but almost all the F1-score of these models close to or above 92 percent. From the perspective of precision and recall, for each model, the difference of these two measurement is limited in around 3.5 percent. In this step, we aim to determine whether the model we trained by our method can afford the task through 5-fold validation. Fortunately, we obtained a series of intelligible result which proves that we can trust our method and valid analysis can be made based on the performance of the spaCy NER model.

5.1.2 Comparison of proprietary model and joint model

After the validation of our training method and performance of the model get verified, we begin to trained proprietary NER model and joint NER model using the cleaned data. For the data in each country, we randomly divide them into training set and test set at the ratio of 7:3, firstly use training set to generate proprietary model for each country while preserve their test set, then merge five training sets together to produce a joint model which is the same as what have been illustrated in **method** part. In each round of training we can get 5 individual models(each country has one), 5 test sets(each country has one) and one joint model. We repeat this procedure five times, so we can have totally 25 of individual models, 25 test sets and 5 joint models on hand. With the purpose of test the performance, in each prediction round, the test set from each country will be used by its corresponding proprietary model as well as the joint model. We also repeat this test procedure five times to provide enough information for research. We group the prediction result by countries, here we take the result from five Australia proprietary NER models and 5 joint models as an example. We use bar plot to display the precision, recall and F1-score of the proprietary models of Australia and joint models.

From the three bar plot which reflect the performance of the individual model and joint model of Australia, whether the metric is precision, recall or F1-score, proprietary models, they all surpass joint models. For any of the proprietary model such as **individual 1**, even its training set is the subset of the **joint model 1**, the precision, recall and F1-score are about 4 to 7 percent higher than the **joint model** (joint model 1 in this case). This result present a fact that on the one hand both individual models

Figure 5.3: Precision result plot of individual models and joint models

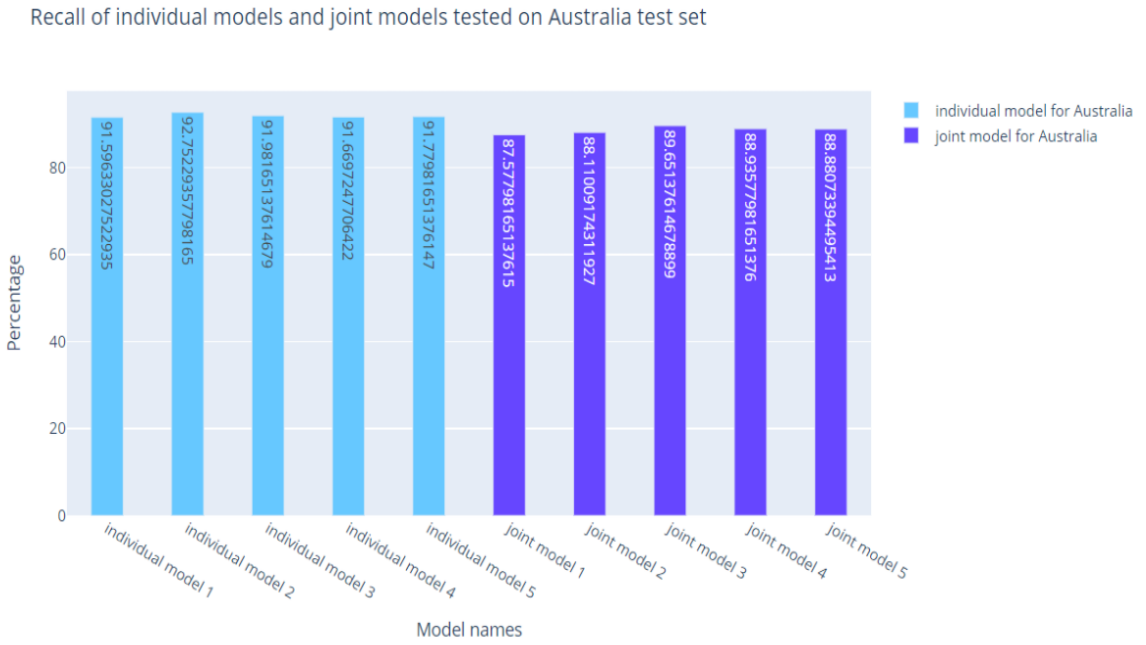

Figure 5.4: Recall result plot of individual models and joint models

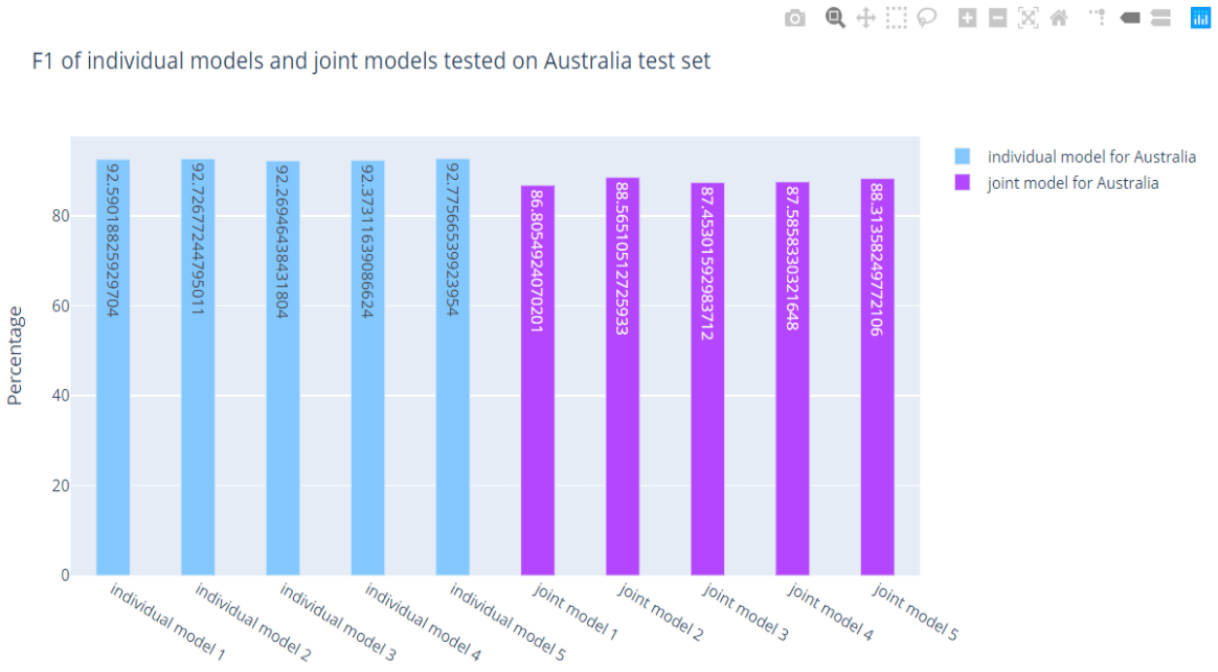

Figure 5.5: F1-score result plot of individual models and joint models

and joint models achieve a comparably good result with about 92 percent in all metrics for individual models and 87 percent in all metrics for joint models, on the other hand, the difference are conspicuous. The proprietary models are more accurate in making prediction on the test sets of its corresponding country than the joint model.

In the light of the result, we propose a hypothesis that because of the numerous training samples(job advertisements) from other countries, the influence of the training sample in Australia is diminished thus the joint models do not perform as good as the proprietary models which use pure Australian training samples. Nonetheless, the joint model may possess the capability to make relatively good prediction in other countries not restricted only in Australia prediction.

The results above are all from the Australia test sets and it is not enough to support us to verify our hypothesis. Therefore, we continue to repeat the predication process on the remaining test sets from other four countries and we concatenate the models' performance and display together in a box plot(we just use F1-score as the metric in Fig 5.6).

By analyzing the value distribution in Fig 5.6, we can find several points:

1. For each country, overall performance of proprietary models still exceed the performance of the joint model.

2. Five joint models make comparatively good predication in test sets from five countries.

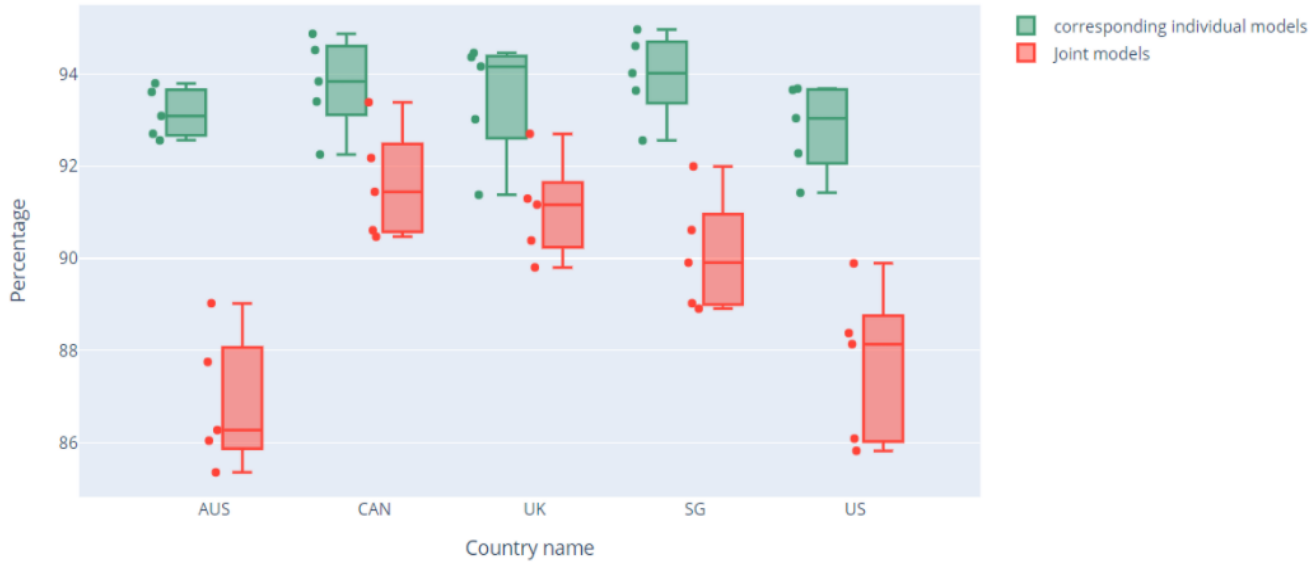

Figure 5.6: F1-score of proprietary models and joint models in different countries

3. The fluctuation in joint model performance is more dramatic than the fluctuation in proprietary model performance.

Further more, we would like to know whether proprietary model for a specific country can make good prediction on the test set from other countries. We fixed Australia proprietary model and test its performance on other four countries' test sets.

Test set	F1-score
Canada test set	77.7
UK test set	74.2
Singapore test set	72.2
US test set	72.4

Table 5.8: Australia model predict on other test sets

We continue to check whether the other four proprietary models have the identical result when they do not predict on their corresponding tests and find their prediction results are agree with the Australia proprietary model.

Through **point 1**,**point 2** and table 5.8, we can corroborate our hypothesis since the joint models trained by the training set consists of five countries job ads do not as efficient as proprietary models for any single country, but they have the ability to be widely deployed in any countries and achieve a not bad prediction performance. To find out the factor that cause **point3**, we need to investigate which factors lead to not correct extraction in our models.

5.1.3 The factor lead to erroneous predication and the performance gap between two kinds of models

In the first step, we check the skill occurrence frequency distribution of each country in that we believe there might exist the frequency of some skills is too large to be ignored by the spaCy training process thus the skewd distribution of the training set produce NER models lacking of comprehensive predication ability. Then we use probability density plot with mean and log probability density plot with mean to show the distribution of the skills in different frequency in the training sets of each country.

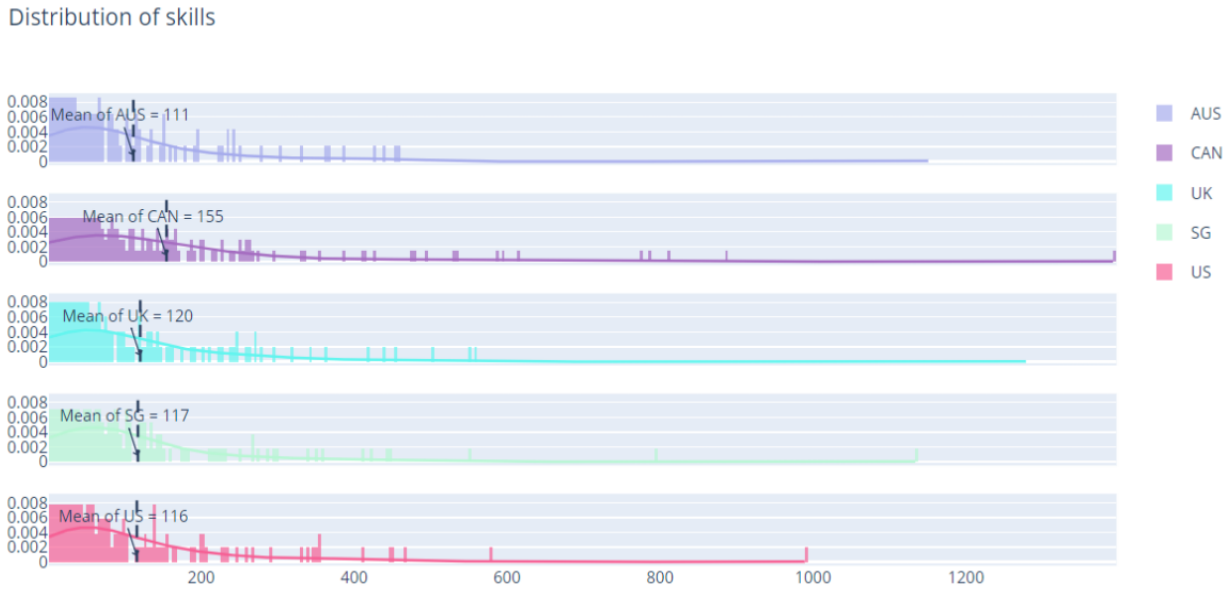

Figure 5.7: Probability density plot of the skills frequency distribution

What is interesting is that the distributions of the frequency of skills in the training set for five countries are quite similar. Large amount of the skills with the frequency lower than 200 form the main part of training set in these five countries. There are also small number of skills with high frequency in the training set of each country. We manually set a threshold 400, and record the skills whose frequency larger than

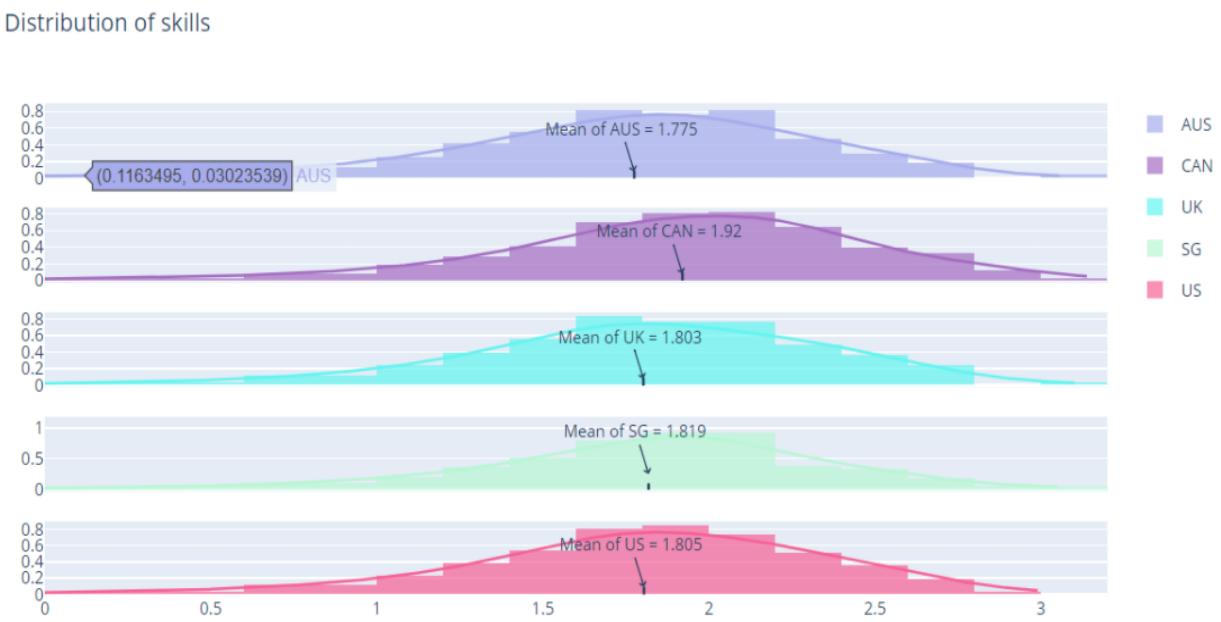

Figure 5.8: Log probability density plot of the skills frequency distribution

400.

Skill name	Occurrence frequency
Communication skills	1153
Teamwork/collaboration	460
Detailed-orientated	453
Attention to detail	437
Planing	426

Table 5.9: High frequency skills in Australia

Skill name	**Occurrence frequency**
Communication skills	1395
Organizational skills	885
Teamwork/collaboration	881
Customer service	785
Detailed-orientated	775
Attention to detail	647
Problem solving	578
Writing	498
Scheduling	474

Table 5.10: High frequency skills in Canada

Skill name	**Occurrence frequency**
Communication skills	1281
Organizational skills	557
Teamwork/collaboration	549
Customer service	504
Detailed-orientated	453
Planing	453
Attention to detail	439

Table 5.11: High frequency skills in UK

Skill name	Occurrence frequency
Communication skills	1136
Teamwork/collaboration	795
interpersonal	549
Planing	447
Team player	441
English	441

Table 5.12: High frequency skills in Singapore

Skill name	Occurrence frequency
Communication skills	991
Detailed-orientated	580
Organization skills	467
Customer service	452
Teamwork/collaboration	440
Writing	416

Table 5.13: High frequency skills in US

When we list the skills which above the threshold with their occurrence frequency, we notice that not only the distribution of skills frequency in each country resembles each other, but the skills with high occurrence frequency are also very alike. Some representative skills like "Communication skills", "Customer service","Teamwork/collaboration" commonly exist in these lists and it is worth noting that these frequent skills universally possess a characteristic that they tend to be general rather than specific in describing the skill. They may bemuse the readers if they just look at the skill names such as "planning", "customer service" or "writing" without context. The skill "planning" can be more detailed like "strategy planning","scenario planning","business planning", the skill "writing" may be specifically divided into "English writing", "hand writing","speech writing" in real life.

In retrospect the process that we scrutinize the skills in the training set from five

countries, we find the skill frequency distribution of these training sets follow the same pattern and the skills with high frequency of each training set do not take dominant position in the whole training set and these skills are roughly the same. This help us eliminate our previous worry about skewed training set decay the performance of the NER model trained by these training sets. For the next step, we compare the prediction result with the groundtruth expecting to discover the factors that contribute to wrong prediction. For the sake of convenience, we separately save the groundtruth data and prediction result in different lists denoted as list **GT** and list **P** for every training sample. By using these two lists, we derive other two lists:

- One is the list of the skills which do not extracted by our NER model, this list consist of the skills only exist in list **GT** but not in **P**, if we regard list **GT** and list **P** as two sets mathematically, this list can be represented as **GT-P**.

- The other is the list of skills which extracted by our NER model but not exist in groundtruth data. Similarly, we denote this list as **P-GT**.

These two derived lists provide information about which skills are tend to be ignored by the NER model and which skills are identified as "new" skills by the NER model. To exploit more points about which skills are less likely to be extracted, we record the skills in list **P-GT** along with the times of this skill is missed in prediction for each training samples of five countries. Like what we did in find out skills with high occurrence frequency, we set 50 as a threshold and record the skills which are missed greater than 50 times.

Skill name	**Times of missed**
Customer service	126
Communication skills	110
Marketing	104
Planning	93
Writing	79
Sales	72

Table 5.14: Missed skills in Australia dataset

Skill name	Times of missed
Customer service	111
Advertising	103
Communication skills	99
Sales	93
Marketing	84

Table 5.15: Missed skills in Canada dataset

Skill name	Times of missed
Communication skills	99
Customer service	86
Designing	81
Research	73
Marketing	64
Sales	63
Writing	55

Table 5.16: Missed skills in UK dataset

Skill name	**Times of missed**
Sales	101
Communication skills	90
Writing	89
Finance	66
Marketing	57
Sales	54

Table 5.17: Missed skills in Singapore dataset

Skill name	**Times of missed**
Management	92
Communication skills	82
Marketing	73
Sales	60
Compounding	51

Table 5.18: Missed skills in US dataset

From these tables, it is not inconspicuous that these skills share a common feature, general. The generality does not only reflect on the omnipresence of most of these skills because in any training set, these skills have high occurrence frequency, but these skills draw a "vague" boundary in describing the skill requirement such as "Research","Designing" can be more specific like what we analyzed before. After we find the NER models are not sensitive to the general skills, we explore the new skills (skills in prediction list which are not in groudtruth list) in the list **P-GT**. As the occurrence of the frequency of each skill in list **P-GT** is smaller than 3 and most of frequency is 1 , then it is not necessary to set a threshold to distinguish skills with low frequency and high frequency. We deliberately select some skills in list **P-GT** of each country to support our illustration of the finding.

Skill name	Occurrence frequency
Problem management	2
Dealer management	1
Digital Marketing	1
System processing	1
Biotech	1
Clinical skills	1

Table 5.19: New skills in Australia dataset

Skill name	Occurrence frequency
Content management	1
Credit management	1
Data management	1
Digital finance	1
Network operation	1
Network design	1
Capacity management	1

Table 5.20: New skills in Canada dataset

Skill name	Occurrence frequency
Project marketing	1
Project planning skills	1
Interpersonal development	1
Customer service management	1
Social media	1

Table 5.21: New skills in UK dataset

Skill name	Occurrence frequency
Problem management	3
Agile management	2
Client care	1
change management	1
Analytical skills	1
English communication skills	1
Plan writing	1
Chinese	1

Table 5.22: New skills in Singapore dataset

Skill name	Occurrence frequency
Excellent customer service	3
Agile management	2
Client care	1
Early childhood education	1
Good analytical skills	1
Italian communication skills	1
Fluent communication	1
Discharge planning	1

Table 5.23: New skills in US dataset

The skills in the tables above are deliberately picked, since from these skills we can find the NER model's prediction tendency. By observing the whole skills in list **P-GT** in five countries, it seems that we now can understand why the skills like "communication skills" are not extracted by the model. These new skills share a common characteristic as well, concrete. Think about the method we used to derive list **GT-P** and list **P-GT**, we treat the ground truth list and prediction list in each training sample as sets in mathematical perspective, and use difference sets to represent list **GT-P** and list **P-GT** in each training sample. Therefore if a element in list **GT** is "Finance", and a element in list **P** is "Digital finance", even though these two skills are nested in text, we consider them different skills. The specificity of these skills is embodied by the NER model will extract the skills that we regard as general in list **GT-P** with a decorated prefix or suffix. The prefix words are most nouns or adjectives while the suffix words are most nouns. As the skill "management" in list **GT-P**, there are many variation of it in list **P-GT**, "Data management","Problem management", "Credit management" in list **P-GT**, other skills like "Marketing", "Network", "Communication skills", "Planning" have more specific version in list **P-GT** as well. Besides the complementary text in the content of the skills, the NER models have the incline to extract the adjective words such as the word "Excellent" in "Excellent communication skills", "Fluent" in "Fluent communication", etc. Although as what we explain, these representative examples are used to help introduce our findings and the amount of the examples seems few, it does not means that they are the only results we have from list **P-GT**. Completely new skills are contained in list **P-GT**, despite the number of new skills is not big, but it proves our model is able to extract skills that it never met.

Not all the general skills in list **GT-P** have specific version in list **P-GT**, some of them are indeed missed by the NER model or lead to useless prediction for example, "protection" in training set make model produce the prediction skill "protection of the" which is meaningless.

Through the our analysis, we now can identified the factors that influence the performance of the NER model are two:

1. The general skills in training set make model hard to learn and lead to meaning less prediction.

2. The tendency of the NER model that predict more specific version of the skill rather than the general version.

5.1.4 Other characteristics of spaCy NER model in prediction

The work of establish NER model to extract the skills in advertisements is done and we are curious about other characteristics of spaCy NER model. Considering the words labelled as entity **SKILL** are mostly noun and the length of the entity is short(usually less than 3 words), we select other entities including "Certification", "Degree", "Email" in order to find out the do attributes of entity affect the prediction. We did the same processing to make the training sets and test sets for these three kinds of entities. For each entity, We trained joint model and test on test set of each country. Here are the result:

Country name	Precision	Recall	F1-score
AUS	67.7	63.5	65.5
CAN	57.5	46.5	51.4
UK	70.5	60.9	65.3
SG	73.3	69.7	71.4
US	48.9	33.8	40

Table 5.24: Performance of the **Certification** NER model

Country name	Precision	Recall	F1-score
AUS	95.7	92.8	94.2
CAN	100	100	100
UK	100	96.1	98.0
SG	78.5	84.6	81.6
US	100	98.9	99.4

Table 5.25: Performance of the **Degree** NER model

Country name	Precision	Recall	F1-score
AUS	99.4	99.0	99.2
CAN	100	99.2	99.6
UK	99.8	99.6	99.7
SG	100	100	100
US	97.8	100	98.9

Table 5.26: Performance of the **Email** NER model

When we check the text which are labelled as "Degree" and "Email", we find they contains indicative word. For "Degree", "Bachelor of", "Bachelor", "Master" and so on can be the indicative word which makes model is able to distinguish easily. The symbol "@" is the indicative word for entity "Email" also reduce the difficulty in model prediction. Because of this, the performance of the NER model when applied to predict "Degree" and "Email" is exciting, "Email" model can reach nearly 100 percent or exactly 100 percent correct in predication and for "Degree" NER model, most of the models achieve roughly 100 percent correct except Australia and Singapore. We explore further why Australia and Singapore do not have as good performance as other countries. Lack of training sample is the main reason, the size of the training set for the other three countries are all bigger than 200, but for Australia, the size of the its training set is 101, for Singapore even worse, it only has 60 data in training set. The test set size also contribute to the even worse performance of the "Degree" NER model in Singapore, there are only 32 data in the test set. The "Degree" model in Singapore predict 14 of them should contains "Degree", 11 out of 14 are correct (Precision= 78.5), while it missed 2 data(Recall=84.6). The situation for the text which labelled as "Certification" is quite different. These text mostly are long sentences consist of 5 words or more without any apparent indicative words. It is hard for model to capture the informative text during training thus result in the performance of "Certification" NER model less accurate than "Degree" and "Email" NER models.

That raise our suspicion about whether the length of the text have influence on the performance of the prediction. Unlike the dearth of the test set in some countries when predict entity "Degree", the test set for "Certification" in each country are sufficient. We then split the test set by the number of words with "Certification" label into five groups in each country . These groups contain the test samples whose length of the annotated text less than or equal to five words, five to ten word, eleven to fifteen word, fifteen to twenty words and more than twenty words respectively. We use these five groups to test our "certification" NER models and try to prove our conjecture.

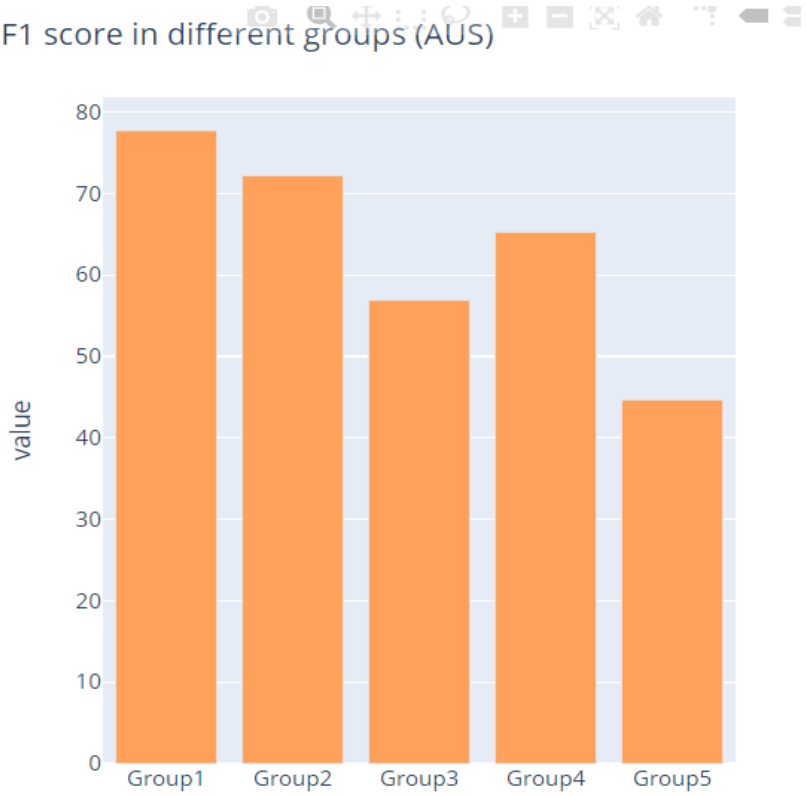

Figure 5.9: F1-score for different groups in "Certification" prediction in Australia

The metric we choose to evaluate the performance of the model is F1-score, and the performance of "certification" models when test on different test groups which contain different length of test samples respectively shows that the length of the test sample truly affect the accuracy of the models. The accuracy of the models tend to decrease when the length of the text become longer at most of time. The ideal accuracy distribution should like Fig 5.13, where obvious accuracy difference can be observed and the accuracy decline rigidly when the length of test sample become longer. However, for most countries, the accuracy does not go all the way down, but have fluctuations when test on their **Group 3** or **Group 4**. Nonetheless, the difference still conspicuous that the accuracy of **Group 1** is the highest among all the groups and it nearly transcends second highest about 10 percent in terms of F1-score in any country. The result approximately accord with our conjecture and it proves that the accuracy indeed tends to decrease when the the length of the text they need to label become longer.

From these prediction result, we find two new factors that influence the prediction accuracy:

1. If the text contains special symbols or indicative words, the NER model will find them easily thus achieve a good performance in prediction.

2. The length of the text that we aim to predict will cause effect on prediction

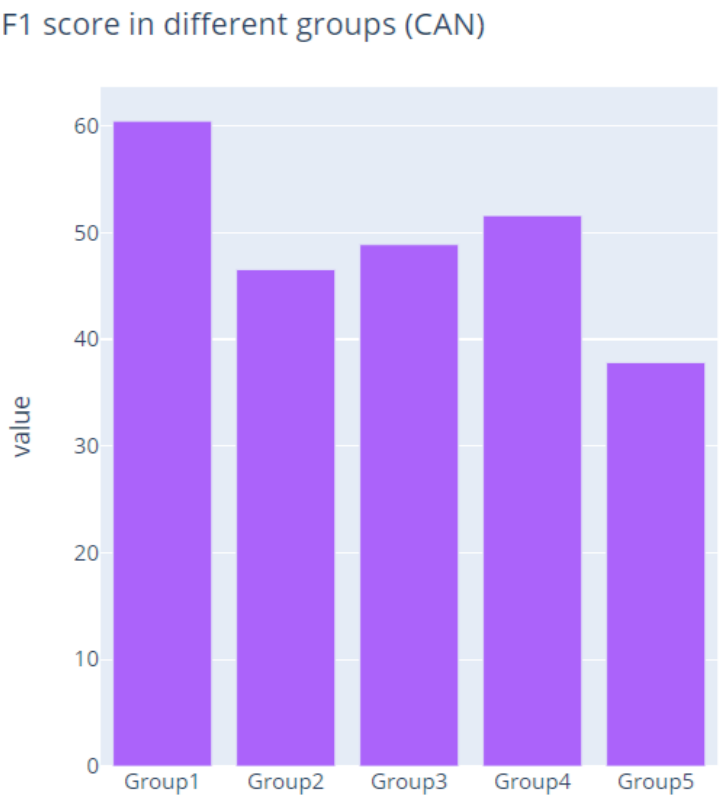

Figure 5.10: F1-score for different groups in "Certification" prediction in Canada

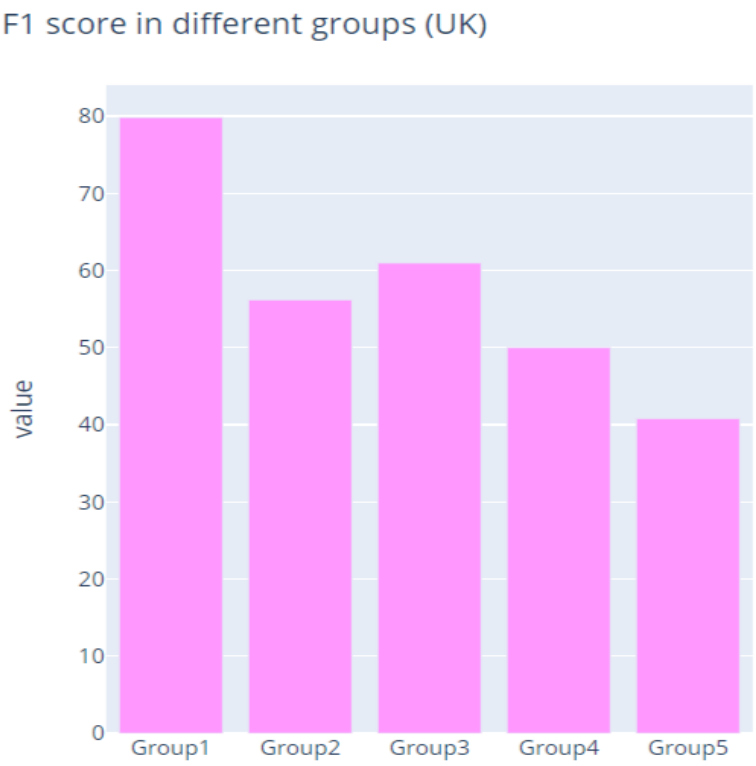

Figure 5.11: F1-score for different groups in "Certification" prediction in UK

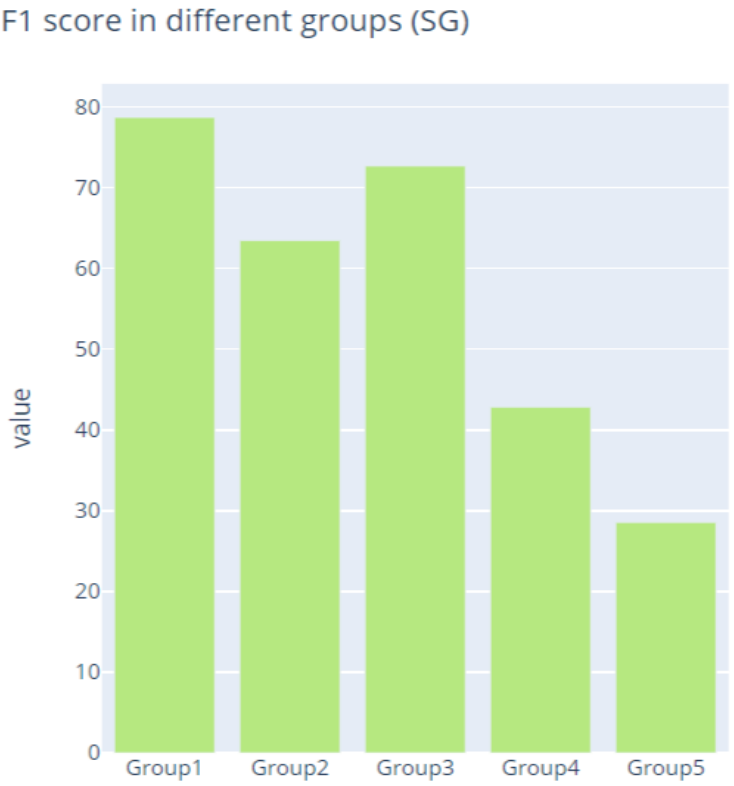

Figure 5.12: F1-score for different groups in "Certification" prediction in Singapore

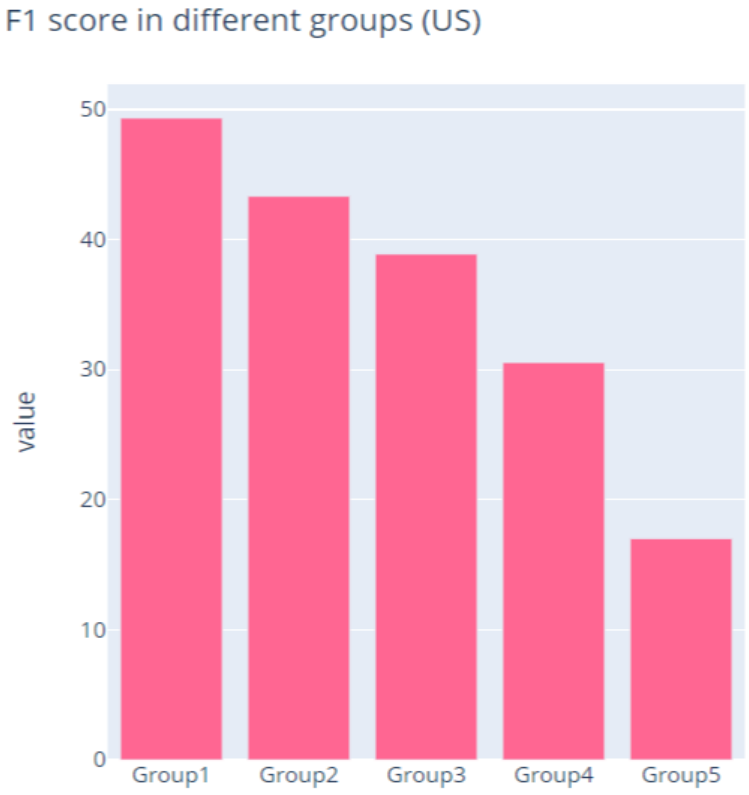

Figure 5.13: F1-score for different groups in "Certification" prediction in US

performance. The longer the text is, the more difficult for the NER model to predict.

5.2 The comparison between spaCy and Flair in NER model training

In the previous part, we analyze the NER model trained by spaCy in the aspect of the performance of prediction and the factors which hurt model performance. In this chapter, we will show the prediction result from the Flair NER model and make comparison with the spaCy model in different aspects.

5.2.1 Training cost comparison

Training a Flair NER model is more intractable than training a spaCy model. From the first step, entity annotation, the workload of annotating a Flair required training set takes more time than annotating a spaCy training set because the Flair training set demand BIO annotation which you need to label every word in each training sample, but the indices annotation in spaCy only need to label the words which are the entity you want to predict.

In addition, Flair training demands more powerful device to support the work, my personal computer can only process the training set with 6000 training samples, and the high performance computer can use 13000 training samples while for spaCy training in our PC we can process 30,000 data in training. The Flair inner structure is more complex than the structure of spaCy means that it takes more time to train a Flair model than train a spaCy one. In addition, our data samples are the job advertisements with long text, it requires large memory when doing word embedding in Flair training.

We finally restrict the size of the training set at 13250 and train 5 joint models using spaCy and Flair respectively then compare the performance of these model when predict on corresponding test set whose size is 1800 for each. We record the time cost and prediction performance for each model.

Before we train the spaCy and Flair models, we made a conjecture that even though

model name	Time cost	F1-score
Joint 1	6h 10min	87.4
Joint 2	6h 18min	88.2
Joint 3	6h 25min	87.9
Joint 4	6h 15min	88.0
Joint 5	6h 31min	88.3

Table 5.27: spaCy NER model cost and effect

the intrinsic complexity may slow down the training speed, but it may enable Flair models perform at least as good as spaCy models. However, from the result above,

model name	Time cost	F1-score
Joint 1	12h 13min	41.4
Joint 2	12h 30min	42.7
Joint 3	13h 01min	43.1
Joint 4	12h 45min	42.9
Joint 5	12h 38min	42.9

Table 5.28: Flair NER model cost and effect

the fact is exact the opposite. Almost two times longer training time does not give Flair models outstanding prediction ability, but make the prediction performance even half accuracy compared with spaCy models.

5.3 Summary

In the chapter, we display our research results of the performance of proprietary models and joint models and analysis the features of these two kinds of model. We find that the proprietary model perform better than joint model when make prediction on a specific country, but joint model can be widely applied to predict more countries skill. In order to find out the reason that may influence the prediction performance, we investigate factors including text peculiarity, length of the entity, indicative words. We find spaCy tend to miss the word with general meaning like "Customer service" and extract more specific version (detailed content or with adjective words) of these general words like "Excellent customer service". The spaCy NER models tend to perform less good when the target text is relatively long. If the text contains some representative words or indicative words, the NER models can find them easily and make good prediction. Despite Flair is recognised as the state-of-the-art NLP framework which offer powerful support to many NLP tasks, it costs too much time and demand high quality hardware to train a NER model but achieve unfavorable result comparing with spaCy in this research.

Chapter 6

Conclusion

At the beginning of this chapter, we will first talk about our primary findings in our research. Next, we introduce the limitation of our work. Then, as our research still have the value for further research, we demonstrate the future work of our research, and finally we discuss the our contribution to the research community, and the give some advice for future researchers who find interests in our research.

6.1 Primary findings

In our research, we are devoted to building a reliable NER model to extract "SKILL" entity, from job advertisements in different English speaking countries. We especially focus on using NLP Framework spaCy to train a qualified NER model and compare it with another type of model trained by NLP framework Flair.

We propose a training strategy that we train two kinds of NER model. The first kind called proprietary model trained by job ads with labelled entities from a specific country, say, each country has their own NER model. The second kind called joint model produced by the combination job ads from five countries. By testing their performance on the test sets from different countries, we find the advantages and disadvantages of these two model. The proprietary model is more suitable to predict skills in the job ads from its corresponding countries since it can achieve nearly 92 percent in F1-score, but less effective when predict skills in the job ads from other countries. The joint model,despite, it can not predict correctly as many as proprietary model can in any countries, it does not constrained by the countries and it can reach 87 percent in F1-score when extract skills in job ads from any country. This result offer us options that according to our goal, we can adpot different model training strategy. If one only concern the skills in a one specific country, the better choice is that he only collect the job ads from this country as training set to train a proprietary model. Similarly, if someone wants to build a model which can be used to extract skills from different countries, you would better use job ads from various countries to train a model.

By researching the factors that leading to erroneous prediction, we find spaCy NER model often misses the skills with general meaning while extracted a complex version of these skills. Some of wrong predictions are not actually wrong, it just extract these

skills with adjective words or words containing supplementary information. SpaCy NER models perform worse when predicting long entity than predicting short entities and its accuracy in predicting entities containing representative words is higher than the accuracy in predicting entities without representative words. These findings tell us we should lower our expectation about the accuracy when we plan to use spaCy NER model to predict entities with long length and we can assume the spaCy model can make good prediction when extract the word with obvious indicative words.

Contrasting the cost and effect using spaCy and using Flair to train a NER model, we find if your devices are not powerful enough or you have time limitation to produce a reliable NER model to predict entity like "SKILL", we recommend that you choose spaCy since the time spent on data processing and training is short than Flair and prediction accuracy is even better.

6.2 Limitation

The work has limitations in the following aspects.

Firstly, our task focus on a specific brunch of NLP and only concentrate on "SKILL" prediction. Out of the strict attitude to this research, we need to point out that these results can be reproduced if followed our research method, but we can not guarantee that when this method is applied to other entities extraction it can achieve similar performance or draw the same conclusion.

Secondly, due to the device limitation, we can not put all the training data we have to train a Flair NER model since the memory will run out if the size of training data larger than 13500 in word embedding. Therefore, we can not know whether the big size of the training set will greatly improve the prediction accuracy when using Flair NER model.

Thirdly, because the complexity and training cost of Flair model, we can not provide solid explanation about what factors contribute to its wrong prediction, and why it has such big difference in prediction accuracy compared with spaCy.

6.3 Contribution to Community and advice for other researchers

Our research provides a tenable method to extract skills from online job ads, as far as we know, there is no precursor who focus on offering a reliable NER model to predict skills in job ads. We are proud to introduce this method and hope this work can be accepted and widely used in predicting other entities in other data source. Still, we can not promise that our method can completely transferred into other NER tasks which aim to predict other entities, but we have confidence that our method will inspire future researchers in some points in some ways . We would suggest the future researchers who find interests in our work and plan to continue extract the skills from the job ads written in English that they should spend time to sifting representative job ads and make sure you have high quality annotation labels. We

also encourage you to try some new NLP framework like **stanza** in this task or refine our Flair training to see whether it can achieve more.

6.4 Future work

It seems that we have finished our job in this research, but it actually yields more work to do in the future.

As what we suggested to future researchers, refining our Flair training strategy, we will also adjust our training method in Flair including recheck annotation problem, compress data size, try other word embeddings. The demand of the online labor market changes periodically, we need to update our scraper and model according to this change for keeping effective. Other than maintain our existing NER model and method, we will explore new approach to solve this task by using new frameworks.

Bibliography

1. Seek about us. In *Retrieved 11 September 2020.* https://www.seek.com.au/about/. (cited on page 11)

2. 2020. xml.etree.elementtree — the elementtree xml api. https://docs.python.org/2/library/xml.etree.elementtree.html#. (cited on page 11)

3. Aboufarw, G. A. M. A., K.S., 2022. Traffic accident risk forecasting using vision transformers,. In *Proc. of the IEEE Intelligent Transport Systems Conference 2022, Macao, China*. (cited on page 7)

4. Akbik, A.; Bergmann, T.; Blythe, D.; Rasul, K.; Schweter, S.; and Vollgraf, R., 2019. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, 54–59. Association for Computational Linguistics, Minneapolis, Minnesota. doi:10.18653/v1/N19-4010. https://www.aclweb.org/anthology/N19-4010. (cited on pages 8, 9, and 20)

5. Akbik, A.; Blythe, D.; and Vollgraf, R., 2018. Contextual string embeddings for sequence labeling. In *COLING 2018, 27th International Conference on Computational Linguistics*, 1638–1649. (cited on page 8)

6. Allen, D. M., 1974. The relationship between variable selection and data agumentation and a method for prediction. *technometrics*, 16, 1 (1974), 125–127. (cited on page 31)

7. Baum, L. E. and Petrie, T., 1966. Statistical inference for probabilistic functions of finite state markov chains. *The annals of mathematical statistics*, 37, 6 (1966), 1554–1563. (cited on page 2)

8. Chiu, J. P. and Nichols, E., 2016. Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics*, 4 (2016), 357–370. doi:10.1162/tacl_a_00104. https://www.aclweb.org/anthology/Q16-1026. (cited on page 2)

9. Colic, N. and Rinaldi, F., 2019. Improving spacy dependency annotation and pos tagging web service using independent ner services. *Genomics & informatics*, 17, 2 (2019). (cited on page 2)

10. contributors, W. Word embedding. https://en.wikipedia.org/wiki/Word_embedding. (cited on page 21)

11. Dawson, N.; Rizoiu, M.-A.; Johnston, B.; and Williams, M. A., 2019. Adaptively selecting occupations to detect skill shortages from online job ads. In *Proceedings - 2019 IEEE International Conference on Big Data, Big Data 2019*, 1637–1643. IEEE, Los Angeles, CA, USA. doi:10.1109/BigData47090.2019.9005967. http://arxiv.org/abs/1911.02302https://ieeexplore.ieee.org/document/9005967/. (cited on page 6)

12. Dr.Michael J.Garbade, 2018. A simple introduction to natural language processing. https://becominghuman.ai/a-simple-introduction-to-natural-language-processing-ea66a1747b3. [Online; accessed 22-Sep-2020]. (cited on page 6)

13. easyai contributors, 2020. Development of named-entity recognition. https://easyai.tech/en/ai-definition/ner/. (cited on page 6)

14. Giannetti, F. Named entity recognition: Challenges and solutions. https://blog.doculayer.com/named-entity-recognition-challenges-and-solutions. (cited on page 19)

15. Goutte, C. and Gaussier, E., 2005. A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. In *European conference on information retrieval*, 345–359. Springer. (cited on page 25)

16. Graves, A.; Jaitly, N.; and Mohamed, A.-r., 2013. Hybrid speech recognition with deep bidirectional lstm. In *2013 IEEE workshop on automatic speech recognition and understanding*, 273–278. IEEE. (cited on page 6)

17. Grigorev, A.; Mihaita, A.-S.; Lee, S.; and Chen, F., 2022. Incident duration prediction using a bi-level machine learning framework with outlier removal and intra–extra joint optimisation. *Transportation Research Part C: Emerging Technologies*, 141 (2022), 103721. doi:https://doi.org/10.1016/j.trc.2022.103721. https://www.sciencedirect.com/science/article/pii/S0968090X22001589. (cited on page 7)

18. Grigorev, M. A. S. K. P. M., A., 2022. Traffic incident duration prediction via a deep learning framework for text description encoding. In *Proc. of the IEEE Intelligent Transport Systems Conference 2022, Macao, China*. (cited on page 7)

19. Honnibal, M. What happens if your annotation has overlapping entity spans? https://support.prodi.gy/t/what-happens-if-your-annotation-has-overlapping-entity-spans/363. (cited on page 17)

20. Honnibal, M. and Montani, I., 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear. (cited on pages 8, 9, 16, 18, and 19)

21. IBM Cloud Education, 2020. What is supervised learning? https://www.ibm.com/cloud/learn/supervised-learning. [Online; accessed 22-Oct-2020]. (cited on page 18)

22. Issa, F.; Monticolo, D.; Gabriel, A.; and Mihăiţă, A., 2014. An intelligent system based on natural language processing to support the brain purge in the creativity process. *IAENG International Conference on Artificial Intelligence and Applications (ICAIA'14) Hong Kong*, (2014). (cited on page 7)

23. Kalchbrenner, N.; Grefenstette, E.; and Blunsom, P., 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 655–665. Association for Computational Linguistics, Baltimore, Maryland. doi:10.3115/v1/P14-1062. https://www.aclweb.org/anthology/P14-1062. (cited on page 6)

24. Kong, Q.; Rizoiu, M.-A.; Wu, S.; and Xie, L., 2018. Will This Video Go Viral: Explaining and Predicting the Popularity of Youtube Videos. In *The Web Conference 2018 - Companion of the World Wide Web Conference, WWW 2018*, 175–178. ACM Press, Lyon, France. doi:10.1145/3184558.3186972. https://arxiv.org/abs/1801.04117http://dl.acm.org/citation.cfm?doid=3184558.3186972. (cited on page 6)

25. Kong, Q.; Rizoiu, M. A.; and Xie, L., 2020. Describing and Predicting Online Items with Reshare Cascades via Dual Mixture Self-exciting Processes. In *International Conference on Information and Knowledge Management, Proceedings*, 645–654. ACM, New York, NY, USA. doi:10.1145/3340531.3411861. https://arxiv.org/pdf/2001.11132.pdfhttps://dl.acm.org/doi/10.1145/3340531.3411861. (cited on page 6)

26. Lafferty, J.; McCallum, A.; and Pereira, F. C., 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. (2001). (cited on page 2)

27. LeCun, Y.; Bengio, Y.; and Hinton, G., 2015. Deep learning. *nature*, 521, 7553 (2015), 436–444. (cited on pages 2 and 7)

28. Lehdonvirta, V., 2018. The rise of online labour markets: freelancing and gig working via internet platforms. (Dec. 2018). (cited on page 1)

29. Li, J.; Sun, A.; Han, J.; and Li, C., 2020. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, (2020). (cited on page 7)

30. Li, X.; Feng, J.; Meng, Y.; Han, Q.; Wu, F.; and Li, J., 2019. A unified mrc framework for named entity recognition. *arXiv preprint arXiv:1910.11476*, (2019). (cited on page 2)

31. Liu, Y.; Liu, Z.; Chua, T.-S.; and Sun, M., 2015. Topical word embeddings. In *Twenty-ninth AAAI conference on artificial intelligence*. Citeseer. (cited on page 21)

32. Mao, M. A. C. C., T., 2019. Traffic signal control optimisation under severe incident conditions using genetic algorithm. *Proc. of ITS World Congress (ITSWC 2019), Singapore*, (Oct. 2019). (cited on page 7)

33. Mao, T.; Mihăită, A.-S.; Chen, F.; and Vu, H. L., 2022. Boosted genetic algorithm using machine learning for traffic control optimization. *Trans. Intell. Transport. Sys.*, 23, 7 (jul 2022), 7112–7141. doi:10.1109/TITS.2021.3066958. https://doi.org/10.1109/TITS.2021.3066958. (cited on page 7)

34. McCallum, A.; Freitag, D.; and Pereira, F. C., 2000. Maximum entropy markov models for information extraction and segmentation. In *Icml*, vol. 17, 591–598. (cited on page 2)

35. Michael Chui, S. L. and Gumbel, P., 2018. How will automation affect jobs, skills, and wages? (Mar. 2018). (cited on page 1)

36. Mihaita, A.-S.; Li, H.; He, Z.; and Rizoiu, M.-A., 2019. Motorway Traffic Flow Prediction using Advanced Deep Learning. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 1683–1690. IEEE, Auckland, New Zealand. doi:10.1109/ITSC.2019.8916852. https://ieeexplore.ieee.org/document/8916852/. (cited on page 6)

37. Mihaita, A.-S.; Liu, Z.; Cai, C.; and Rizoiu, M.-A., 2019. Arterial incident duration prediction using a bi-level framework of extreme gradient-tree boosting. In *Proceedings of the 26th ITS World Congress*, 1–12. Singapore. http://arxiv.org/abs/1905.12254. (cited on page 6)

38. Mihaita, A.-S.; Papachatgis, Z.; and Rizoiu, M.-A., 2020. Graph modelling approaches for motorway traffic flow prediction. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)* (Rhodes, 2020), 1–8. IEEE Press. doi:10.1109/ITSC45102.2020.9294744. https://doi.org/10.1109/ITSC45102.2020.9294744. (cited on page 7)

39. Mihaita, L. H. R. M., A.S., 2020. Traffic congestion anomaly detection and prediction using deep learning. doi:arXiv:2006.13215. (cited on page 7)

40. Mihaita A. S., C. O. C. M. C. C., Dupont L., 2018. Air quality monitoring using stationary versus mobile sensing units: a case study from lorraine, france. *Proc. of ITS World Congress (ITSWC 2018), Copenhagen, Denmark*, (Sep. 2018).

41. Mihaita A. S., M. C. C. C., Benavides, 2019. Predicting air quality by integrating a mesoscopic traffic simulation model and air pollutant estimation models. *International Journal of Intelligent Transportation System Research (IJITSR)*, 17, 2 (2019), 125–141. doi:DOI:10.1007/s13177-018-0160-z. https://link.springer.com/article/10.1007/s13177-018-0160-z.

42. Mihăiţă, A.; Camargo, M.; and Lhoste, P., 2014. Evaluating the impact of the traffic reconfiguration of a complex urban intersection. *10th International Conference on Modelling, Optimization and Simulation (MOSIM 2014), Nancy, France, 5-7 November 2014*, (Nov. 2014).

43. Mihăiţă, A. S.; Tyler, P.; Menon, A.; Wen, T.; Ou, Y.; Cai, C.; and Chen, F., 2017. An investigation of positioning accuracy transmitted by connected heavy vehicles using dsrc. *Transportation Research Board - 96th Annual Meeting, Washington, D.C.*, (Jan. 2017).

44. Mihăiţă, S. and Mocanu, S., 2011. An energy model for event-based control of a switched integrator. *IFAC Proceedings Volumes*, 44, 1 (2011), 2413–2418. doi: https://doi.org/10.3182/20110828-6-IT-1002.02082. https://www.sciencedirect.com/science/article/pii/S1474667016439741. 18th IFAC World Congress. (cited on page 7)

45. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J., 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26 (2013), 3111–3119. (cited on page 21)

46. Mishra, S.; Rizoiu, M.-A.; and Xie, L., 2018. Modeling Popularity in Asynchronous Social Media Streams with Recurrent Neural Networks. In *International AAAI Conference on Web and Social Media (ICWSM '18)*, 1–10. Stanford, CA, USA. https://arxiv.org/pdf/1804.02101.pdf. (cited on page 6)

47. Misra, I.; Zitnick, C. L.; and Hebert, M., 2016. Shuffle and learn: unsupervised learning using temporal order verification. In *European Conference on Computer Vision*, 527–544. Springer. (cited on page 19)

48. Mitkov, R.; Orasan, C.; and Evans, R., 1999. The importance of annotated corpora for nlp: the cases of anaphora resolution and clause splitting. In *Proceedings of Corpora and NLP: Reflecting on Methodology Workshop*, 1–10. Citeseer. (cited on page 2)

49. Monticolo, D. and Mihăiţă, A., 2014. A multi agent system to manage ideas during collaborative creativity workshops. *International Journal of Future Computer and Communication (IJFCC)*, 3, 1 (Feb. 2014), 66–70. doi:10.7763/IJFCC.2014.V3.269. (cited on page 7)

50. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Kopf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S., 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32* (Eds. H. Wallach; H. Larochelle; A. Beygelzimer; F. d'Alché-Buc; E. Fox; and R. Garnett), 8024–8035. Curran Associates, Inc. http://papers.neurips.cc/paper/

9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf. (cited on page 8)

51. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; and Duchesnay, E., 2011. Scikit-learn: Machine learning in Python:log$_l$oss. *Journal of Machine Learning Research*, 12(2011), 2825 − −2830. (*citedonpage*18)

52. Pennington, J.; Socher, R.; and Manning, C. D., 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543. (cited on page 21)

53. Powers, D. M., 2020. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*, (2020). (cited on page 23)

54. Rachel Lea Ballantyne Draelos, 2020. Measuring performance: The confusion matrix. https://glassboxmedicine.com/2019/02/17/measuring-performance-the-confusion-matrix/. [Online; accessed 17-Dec-2020]. (cited on page 24)

55. Rai, A. Nlp | iob tags. https://www.geeksforgeeks.org/nlp-iob-tags/. (cited on page 13)

56. Ramshaw, L. A. and Marcus, M. P., 1995. Text chunking using transformation-based learning. *CoRR*, cmp-lg/9505040 (1995). http://arxiv.org/abs/cmp-lg/9505040. (cited on page 14)

57. Ratinov, L. and Roth, D., 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, 147–155. (cited on page 19)

58. Rizoiu, M. A. and Velcin, J., 2011. Topic extraction for ontology learning. In *Ontology Learning and Knowledge Discovery Using the Web: Challenges and Recent Advances* (Eds. W. Wong; W. Liu; and M. Bennamoun), 38–60. IGI Global. ISBN 9781609606251. doi:10.4018/978-1-60960-625-1.ch003. http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-60960-625-1.ch003. (cited on page 6)

59. Rizoiu, M.-A. and Xie, L., 2017. Online Popularity under Promotion: Viral Potential, Forecasting, and the Economics of Time. In *International AAAI Conference on Web and Social Media (ICWSM '17)*, 182–191. Montréal, Québec, Canada. https://aaai.org/ocs/index.php/ICWSM/ICWSM17/paper/view/15553https://arxiv.org/pdf/1703.01012.pdf.

60. Rizoiu, M. A.; Xie, L.; Caetano, T.; and Cebrian, M., 2016. Evolution of privacy loss in wikipedia. In *WSDM 2016 - Proceedings of*

the 9th ACM International Conference on Web Search and Data Mining, 215–224. ACM, ACM Press, New York, New York, USA. doi:10.1145/2835776.2835798. http://dl.acm.org/citation.cfm?doid=2835776.2835798http://arxiv.org/abs/1512.03523http://dx.doi.org/10.1145/2835776.2835798. (cited on page 6)

61. Santra, A. and Christy, C. J., 2012. Genetic algorithm and confusion matrix for document clustering. *International Journal of Computer Science Issues (IJCSI)*, 9, 1 (2012), 322. (cited on page 23)

62. Sapphire Duffy, 2020. Is flair a suitable alternative to spacy? https://medium.com/@sapphireduffy/is-flair-a-suitable-alternative-to-spacy-6f55192bfb01. [Online; posted 27-January-2020]. (cited on page 8)

63. Shaffiei, A. C. C., S. Mihaita, 2019. Demand estimation and prediction for short-term traffic forecasting in existence of non-recurrent incidents. *Proc. of ITS World Congress (ITSWC 2019), Singapore*, (Oct. 2019). (cited on page 7)

64. Shafiei, M. A. N. H. B. C. D. B. C. C., Sajjad, 2020. Short-term traffic prediction under non-recurrent incident conditions integrating data-driven models and traffic simulation. In *Transportation Research Board (TRB) 99th Annual Meeting, Washington D.C.* doi:http://hdl.handle.net/10453/138721. (cited on page 7)

65. Shafiei, S.; Mihăiţă, A.-S.; Nguyen, H.; and Cai, C., 2022. Integrating data-driven and simulation models to predict traffic state affected by road incidents. *Transportation Letters*, 14, 6 (2022), 629–639. doi:10.1080/19427867.2021.1916284. https://doi.org/10.1080/19427867.2021.1916284. (cited on page 7)

66. Stehman, S. V., 1997. Selecting and interpreting measures of thematic classification accuracy. *Remote sensing of Environment*, 62, 1 (1997), 77–89. (cited on page 23)

67. Unwin, J. T.; Routledge, I.; Flaxman, S.; Rizoiu, M. A.; Lai, S.; Cohen, J.; Weiss, D. J.; Mishra, S.; and Bhatt, S., 2021. Using hawkes processes to model imported and local malaria cases in near-elimination settings. *PLoS Computational Biology*, 17, 4 (apr 2021), e1008830. doi:10.1371/JOURNAL.PCBI.1008830. http://medrxiv.org/content/early/2020/07/17/2020.07.17.20156174.abstracthttps://dx.plos.org/10.1371/journal.pcbi.1008830. (cited on page 6)

68. Wang, T.; Huan, J.; and Li, B., 2018. Data dropout: Optimizing training data for convolutional neural networks. In *2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*, 39–46. IEEE. (cited on page 20)

69. Wen, T.; Mihăiţă, A.-S.; Nguyen, H.; Cai, C.; and Chen, F., 2018. Integrated incident decision-support using traffic simulation and data-driven models. *Transportation Research Record*, 2672, 42 (2018), 247–256. doi:10.1177/0361198118782270. https://doi.org/10.1177/0361198118782270. (cited on page 7)

70. Wikipedia contributors, 2020. Confusion matrix. https://en.wikipedia.org/wiki/Confusion_matrix#cite_note-7. [Online; accessed 16-Dec-2020]. (cited on page 23)

71. Wikipedia contributors, 2020. Inside–outside–beginning (tagging). https://en.wikipedia.org/wiki/Inside_outside_beginning_(tagging)#cite_note-2. [Online; accessed 20-Sep-2020]. (cited on page 14)

72. Wikipedia contributors, 2020. Loss function. https://en.wikipedia.org/wiki/Loss_function. [Online; accessed 26-Oct-2020]. (cited on page 18)

73. Wikipedia contributors, 2020. Loss function for classification. https://en.wikipedia.org/wiki/Loss_functions_for_classification#Logistic_loss. [Online; accessed 26-Nov-2020]. (cited on page 18)

74. Wikipedia contributors, 2020. Named-entity recognition. https://en.wikipedia.org/wiki/Named-entity_recognition. [Online; accessed 22-Sep-2020]. (cited on page 6)

75. Wikipedia contributors, 2020. Natural language processing. https://en.wikipedia.org/wiki/Natural_language_processing. [Online; accessed 22-Sep-2020]. (cited on page 6)

76. Wikipedia contributors, 2020. Precision and recall. https://en.wikipedia.org/wiki/Precision_and_recall#Precision. [Online; accessed 18-Dec-2020]. (cited on page 25)

77. Wu, S.; Rizoiu, M.-A.; and Xie, L., 2019. Estimating Attention Flow in Online Video Networks. *Proceedings of the ACM on Human-Computer Interaction*, 3, CSCW (nov 2019), 1–25. doi:10.1145/3359285. http://dl.acm.org/citation.cfm?doid=3371885.3359285. (cited on page 6)

78. Wu, S.; Rizoiu, M. A.; and Xie, L., 2020. Variation across scales: Measurement fidelity under Twitter data sampling. In *Proceedings of the 14th International AAAI Conference on Web and Social Media, ICWSM 2020*, 715–725. https://arxiv.org/abs/2003.09557.

79. Zhang, R.; Walder, C.; and Rizoiu, M.-A., 2020. Variational Inference for Sparse Gaussian Process Modulated Hawkes Process. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34, 04 (apr 2020), 6803–6810. doi:10.1609/aaai.v34i04.6160. http://arxiv.org/abs/1905.10496https://aaai.org/ojs/index.php/AAAI/article/view/6160. (cited on page 6)

80. Zhao, M. A. O. Y. S. S. G. H. Q. K. T. G. L. M., D., 2022. Real-time attention-augumented spatio-temporal networks for video-based driver activity recognition. In *Proc. of the IEEE Intelligent Transport Systems Conference 2022, Macao, China*. (cited on page 7)