Learning to Answer Recruitment Application Forms

Kah Siong, Tan

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Master of Web Science and Big Data Analytics

of

University College London.

Department of Computer Science

University College London

September 8, 2017

This report is submitted as part requirement for the MSc in Web Science and Big Data Analytics at University College London. It is substantially the result of my own work except where explicitly indicated in the text. The report will be distributed to the internal and external examiners, but thereafter may not be copied or distributed except with permission from the author.

Abstract

Can common questions asked on recruitment application forms be answered from the content of the CVs without supervision? Fully understanding the content of CVs by a machine requires the ability to first read the CVs and then be able to answer questions pertaining to the CV content. Existing work on Question and Answering focused on Machine Reading and Community Question and Answer. In recruitment domain, work based on entity extraction and domain ontologies has been done. In this paper three different approaches were formulated to automatically answer four recruitment application questions with content of unlabeled CVs. A novel Heuristic Evaluation was also implemented to evaluate the relative performance of different algorithms without the need for labeled CVs. A proof of concept software was implemented to illustrate the answering capability.

Acknowledgements

I would like to thank Dr. Jun Wang, my academic supervisor, for his inputs and invaluable advice especially in setting the direction of this MSc project. I would also like to thank Dr. Emine Yilmaz and UCL Ph.D student Rishabh Mehrotra for their advice at major checkpoints of the project.

I would like to thank Mr. Charles Hipps, my industrial supervisor for the rare opportunity to work on actual production data. I would also like to thank Jack Hobson for his wide range of support in this project. Last but not least, i must thank Richard V Poulten for his fantastic technical support, and Charlotte Szostek and Ben Smithers for their strong academic viewpoints.

Contents

1 Introduction **17**

2 Related Work **19**

2.1 Machine Reading Comprehension 19

2.2 Word embedding 20

2.3 Sentence embedding 21

3 Data analysis and insights **22**

3.1 Data introduction 22

3.1.1 Data Nomenclature 22

3.1.2 Data Architecture 22

3.2 Insights 27

3.2.1 Semantic Similarity 27

3.2.2 Data retrieval 27

3.2.3 Language Disparity 27

3.2.4 CV Splitting 27

4 Proposed Approaches **28**

4.1 Problem formulation 28

4.2 Inverse Answering by Similarity (SIM) 30

4.3 Inverse Answering by Classification 31

4.4 Inverse Answering by Topic Modelling (TOPIC) 33

4.5 Vector Representation Models 34

4.5.1 Term Frequency (TF) 34

4.5.2 Term Frequency Inverse Document Frequency (TFIDF) . . 35
4.5.3 Word2vec(W2V) 35
4.5.4 Paragraph2Vec (D2V) 36
4.6 Similarity Measure 36

5 Experiments **37**
5.1 Objectives 37
5.1.1 Compare vector representation models 37
5.1.2 Compare proposed approaches 38
5.1.3 Classification performance 38
5.1.4 Assess evaluation approaches 38
5.1.5 Effectiveness across domains 38
5.1.6 Optimization 38
5.2 Setting up the data 38
5.2.1 Application forms 39
5.2.2 CVs 39
5.3 Preprocessing of data 42
5.4 Setting up for evaluation 42
5.4.1 Crowd-sourced evaluation 42
5.4.2 Heuristic evaluation 44
5.5 Selecting performance metrics 45
5.5.1 Accuracy 45
5.5.2 Multi-Class F1 46
5.5.3 Significance Testing of results 46
5.6 Results 47
5.6.1 Comparison of Vector Representation Model 47
5.6.2 Comparison of approaches 51
5.6.3 Classification Performance 55
5.6.4 Robustness of Heuristic Evaluation 57
5.6.5 Cross-domain effectiveness 59
5.6.6 Optimization 61

6 Conclusion **64**

6.1 Future work 65

Appendices **66**

7 Appendices **66**

A Performance Metrics **67**

A.1 Accuracy 67

A.2 Precision, Recall and F1 67

A.3 Multi-class Precision, Recall and F1 68

A.4 Generalized McNemar's Test 69

B Sample graduate CV **72**

C Sample lateral CV **73**

D Source codes **74**

Bibliography **75**

List of Figures

3.1 Architecture of data set 23

3.2 A distribution on the industries for the 89 clients in the dataset. The size of the circles is directly proportional to the number of clients. The government sector takes up the largest portion of the dataset, with the finance sector coming up second largest. 24

3.3 Typical structure of an Application. *pagetitle* represents broad questions such as 'Education' and 'Work Experience'. Typical key-value pairs reflect specific questions and answers corresponding to the broad question 24

3.4 Typical structure of an Opportunity form 26

3.5 The distribution of file formats for the CVs in the dataset. The largest set of CVs comes in the form of PDF, DOCX and DOC formats. 26

4.1 A realistic representation of the problem formulation. (a) Applications have broad questions that needs to be answered. (b) CVs are broken down into sections. (c) Sections of an applicant's CV will be used to answer the broad question. Content of Applications and CVs are largely different in terms of format and language structures, giving this task an added challenge. 29

4.2 Overview of Inverse Answering by Similarity approach. The vector representation model would be built using a training set of CV Sections (Not shown in figure). The prediction model would compare the similarity between the items in the Applications and the unseen CV section before arriving at the most likely broad question for the CV section. 30

4.3 Overview of Inverse Answering by Classification approach. The vector representation model would be built using the training set of CV sections (Not shown in figure). The prediction model would be trained using the specific questions and answers from Application. It would then predict the most likely broad question for the CV section. 32

4.4 Overview of the Inverse Answering by Topic Modelling approach. A topic model of the entire training set of CV sections is first built. Topics are manually mapped to corresponding broad questions. The topic-word distribution of each unseen CV can now be inferred and thus a prediction be made on the CV section's broad question. . . . 33

5.1 Objectives are to identify suitable vector representation and prediction models,then to answer questions such as classification performances, evaluation approaches and cross domain reliability. Last objective is to evaluate effects of data preprocessing on the proposed approaches. 37

5.2 Results of the LDA model applied onto the test set of CV sections. The results showed that the documents can be be broken down into four topics and demonstrates the validity of the sectioning of the CVs. 41

5.3 A screenshot of the implemented labelling software with some data masked to protect applicant's privacy. Each CV section was displayed in blue while the main instructions were displayed in green. After confirming the answer, the next CV section would be displayed to the user for labelling. 43

5.4 Understanding Heuristic Evaluation. Every CV comes with an Application. This means we predicted a CV section to answer to 'Personal Details', we should find exact words in the corresponding answers of the broad question in its Application. If a certain number of non-stopwords such as names (highlighted in red) appears in both Application answers and CV section, take it to be a HIT, else it would be a MISS. 45

5.5 Pairwise experiments between various vector representations of predefined vector dimensions with various prediction models. The Inverse Answering by Classification approach experimented with Logistic Regression(LOG), Support Vector Machine (SVM), Multi-Layer Perceptron(MLP), Boosted Tree (XGB) and Naive Bayes (NB) algorithms. Topic modelling is the only model that will use the full vocabulary of the training corpus. 48

5.6 W2V vector representation model performed better across all approaches. D2V was consistently poor across all models. Its performance on various classifiers was compared to that of a dummy classifier which performs classification at random, the results suggested that D2V cannot represent the datasets in the experiments. The Inverse Answering by Similarity approach performed consistently better with all vector representation models compared to other approaches 49

5.7 The rows of confusion matrices are the actual values while the columns are the predicted values. The classes are in order of 'Education', 'Skills', 'Personal Details' and 'Experience'. Darker cell indicates a higher number. 'Personal Details' class was best classified for all approaches. The similarity and topic modeling approaches are more consistent with the classification across classes. The Inverse Answering by Classification approach's classifiers had problems with 'Experience' class, frequently misclassifying it as 'Personal Details'. 56

5.8 Heuristic Evaluation evaluate by seeking identical words between the CV Sections and their corresponding Applications, if these number of words exceeds a threshold, it will be a HIT, else a MISS. Thus, Heuristic Evaluation can only be evaluated via accuracy. The figure shows the accuracy results of both Heuristic and the gold standard Crowdsourced Evaluation. Heuristic Evaluation is not representative of the performance of each approach; however, a visual inspection shows they are representative of the relative performance of each approach. 58

5.9 Comparison between results when tested with different domains and different variety of CVs. When tested in a different industry/domain (E.g. Finance to Real Estate), all approaches either perform better or have very little drop in performance. When tested on a wide variety of CVs, all but Inverse Answering by Similarity approach suffered a significant drop in performance. 59

5.10 Confusion matrices of the results when using a lateral variety of CVs. For similarity model, its slight performance drop was mainly due to the 'Work Experience' class. All approaches held well for the 'Personal Details' class but have varying results for the rest of the classes. 60

5.11 F1 scores of each approach before and after the data flattening and lemmatization. There were no strong impact on TOPIC and SIM models but improve SVM and LOG slightly. MLP suffered a slight dip in performance. 62

B.1 Sample of graduate CV broken into sections by the splitting algorithm, numbered 1 to 12 72

C.1 Sample of lateral CV broken into sections by the splitting algorithm, numbered 1 to 3. 73

List of Tables

2.1 Comparison of different approaches used in related domains 21

5.1 Analyzing the top 20 most relevant words for each topic in the generated topic model by LDA. By observation, topics 1,2,3,4 were identified as 'Work Experience','School Extra Curricular Activities','Education','Skills' and 'Personal Details' respectively. 42

5.2 Dimension (dim) size refers to the size of each feature vector, this contributes directly to the feature size of 150,000 samples. Sparsity of 0 means there are only non-zero elements in vector samples. Model time shows time needed to load representation model for inference. Inference time refers to the time taken to infer 100 test documents 49

5.3 An example of how well each vector model can represent 2 similar CV sentences that were both labeled as 'Education'. The sentences were converted to lower case, stripped of stop words and punctuation before being represented by each vector model. A cosine similarity score between the sentences were computed for each vector model. W2V performed much better than the baselines TF and TFIDF. Surprisingly, D2V performed very badly as well. 50

5.4 An illustrated example of the vector model's effect on the SIM approach. Given a sample graduate CV and its numbered sections portrayed in Appendix B, the sections selected to answer the four questions using Inverse Answering by Similarity approach with TFIDF, W2V and D2V vector models are displayed. The table shows that with W2V, most of the answers make sense to humans. TFIDF did not do as well and D2V's answers are not comprehensible. Note that a human could easily relate sections 1 and 12 to answer 'Personal Details', section 2 to answer 'Skills', sections 3 to 6 to answer 'Education' and sections 7 to 11 to answer 'Work Experience'. . . . 52

5.5 F1 scores of the Inverse Answering by Similarity approach's prediction model on W2V, and Topic modelling based on TF with vocabulary size of approximately 500,000. The Inverse Answering by Topic Modelling approach appeared to perform slightly better than the Inverse Answering by Similarity approach. This was however deemed inconclusive after significance testing. 52

5.6 Was the Inverse Answering by Topic Modelling approach performing better than the Inverse Answering by Similarity approach by chance? Stuart Maxwell Test was performed here. A p-value of less than 0.05 rejects the null hypothesis that the results of the approaches are statistically not different. In this case, the p-value of 0.11 was much above 0.05 and thus it cannot reject the null hypothesis. Therefore, the better performance of one approach over the other was not conclusive 52

5.7 Supervised Learning Classifiers and their parameters used in the Inverse Answering by Classification approach. 53

5.8 F1 scores of Inverse Answering by Classification approach based on W2V vector models. LOG, SVM and MLP are the only classifiers that can perform at 0.7 and above, with LOG and SVM taking the lead. NB and XGB performed badly. 53

5.9 P-values based on Stuart Maxwell Test or Wilcoxon's Sign Test if the former fails. A P-value of less than 0.05 rejects the null hypothesis that there is no significant difference between the results of the approaches. In this case, the pairwise performance of each classifier were statistically different. 54

5.10 The Inverse Answering by Similarity approach do not require any form of training prior to prediction but it took extended time to load the training set for cosine similarity computation at prediction later. The Inverse Answering by Topic Modelling and Inverse Answering by Classification approaches used CV Sections and Applications for training respectively but the loading time was much lower than the Inverse Answering by Similarity approach. In terms of prediction time, the Inverse Answering by Similarity approach incurred longer delays due to cosine similarity matrix computation 54

5.11 F1 score of each class for each approach of choice. The approaches performed best on the 'Personal Details' class while the classification of 'Work Experience' class was the least performing among the classes. Despite that, all the classification achieves at least an F1 score of 0.70. 56

5.12 An illustrated example on how a less structured CV affected the results. Given a sample lateral CV and its numbered sections portrayed in Appendix C, the sections selected to answer the four questions using Inverse Answering by Similarity approach with TFIDF, W2V and D2V vector models are displayed. In this particular sample, it shows that the usually accurate 'Personal details' was predicted wrongly as 'Work Experience'. It should be noted that logically, section 1 would answer 'Personal Details', section 2 would answer 'Work Experience' and section 3 would answer 'Education'. 61

5.13 The top 20 most relevant words for each topic generated by LDA. By observation, topics 1,2,3,4,5 are identified as 'Work Experience, School Extra Curricular Activities (ECA)', 'Education', 'Personal Details' and 'Skills' respectively. Since ECA is considered schooling, CV sections that had highest probability for topic 2 were classified into 'Education' class. 63

A.1 Binary class confusion matrix. The rows represent the gold labels while the columns represent the predicted classes 67

A.2 Multi-class Confusion Matrix. The rows represent the gold labels while the columns represent the predicted classes 68

A.3 4x4 Agreement table for McNemar's test.A, B, C and D denoted Education, Skills, Personal Details and Work Experience respectively 69

Chapter 1

Introduction

More employers are turning to online recruitment for their hiring needs. To attract the best candidates, attention has shifted from an employer centric recruitment towards a candidate oriented experience. For the recruiters, the abundance of data provides them the opportunity of building and exploiting insights. In a hiring scenario, despite having to upload CVs into the application forms, candidates are still required to fill in information that are already available in the CVs. Can some of the common questions on the application forms be answered from the content of the CVs without supervision?

In this paper, we formulated three different approaches, each answering four broad recruitment application questions based on content of unlabeled CVs. The Inverse Answering by Similarity approach compute the content similarity between candidate filled Application and uploaded CVs in predicting the questions that each CV section may answer to. The Inverse Answering by Topic Modelling is an unsupervised approach that harness the topic-word distribution of CVs to predict their association to the broad question on the Applications. The Inverse Answering by Classification approach uses the candidate's accompanying Application to perform a supervised learning classification.

The proposed approaches were evaluated via Crowdsourced Evaluation, where subject matter experts were asked to label CV sections to its corresponding broad question. A labeling software was designed and implemented to support the Crowdsourced Evaluation. A novel Heuristic Evaluation that evaluates the relative perfor-

mance of different algorithms without the need for labeled CVs was also proposed and implemented.

The Inverse Answering by Topic Modelling is the most effective approach among the three proposed approaches. It's independence from application forms and efficient running times makes it the most appealing approach in practical scenarios. The Inverse Answering by Similarity approach when used with the Word2Vec vector representation model, is on par with the Inverse Answering by Topic Modelling approach but its current implementation is not efficient enough for practical applications. Logistic Regression remains the better performing of the algorithms experimented in the Inverse Answering by Classification approach. Although its performance is less than that of the Inverse Answering by Similarity approach, its efficiency in the execution of the whole pipeline makes it a more practical approach to use. The Heuristic Evaluation was found to be very reliable in assessing the relative performance of different algorithms, but is not suitable for evaluating individual algorithms. This gave the Recruitment Agency the capability to perform large scale comparison of algorithms without the need for expensive labeling.

In support of this work, approximately 12000 lines of Python, R and bash code in 77 modules were developed with notable library support from Scikit-Learn[1], Keras[2] and Coin[3] among other lesser libraries. The source codes are available in GitHub[4].

Chapter 2

Related Work

2.1 Machine Reading Comprehension

Academic research in the domain of understanding CVs has been limited, potentially due to the sparsity of publicly available CVs. Domain ontology[5, 6], taxonomy[7] and Named Entity Extraction[8] were frequently exploited for such tasks. Named Entity Extraction is a well-established Natural Language Processing (NLP) technique but it is necessary to maintain a form of domain data dictionary to map the extracted entities to the semantic categories. Before work can be done with domain ontologies or taxonomies, the ontologies or taxonomies must first be made available either by building it, or reusing relevant ontologies such as the ResumeRDF Ontology Specification[9]. Tosik et al[10] and Zhang et al[11] focused on known segmented CV sections or semi-structured CV content to extract information with semantic properties. CVs comes in all size and shapes; thus, it is not always possible to arrive at sufficient segmented sections. Commercial products such as TextKernel[12], Sovren[13] and Daxtra[14] developed CV parsers through their expertise in the recruitment domain with technologies such as machine reading, machine learning, taxonomies and ontologies, details of their implementation was not made publicly available though.

Fully understanding the content of CVs by a machine requires the ability to first read the CVs and then answer questions pertaining to the CV content. Generalizing, this field of research is termed as Machine Reading Comprehension (RC).

Work in RC has progressed with the availability of human annotated data sets such as Rajpurkar et al[15], Chen et al[16] and REMEDIA[17], giving rise to one of the earliest Supervised Learning approach by Ng et al[18]. Recent work by Cui et al[19] and Hermann et al[20] followed the advancement of deep learning and incorporated neural networks in the comprehension of text. Another similar branch of task, Community Question and Answering (CQA), involves finding ranked answers to new questions posted by the community. Typical CQA systems such as Freebase[21] and YahooAnswers[22] held databases of community supplied questions and voted answers, which form extensive data sets that can be used for new research. Malhas et al[23] proposed a Learning To Rank model over the labelled questions and answers, using language models and word embedding as features, ranking potential answers to a community question. Zhou et al[24] proposed representing the questions with a variable-cardinality word embedding vector and then generating fisher vectors with a generative model in the Fisher Kernel framework. Question retrieval is performed by calculating the similarity between the fisher vectors of a queried and existing question. Both approaches used word embedding for part of their work.

Various approaches of RC have been proposed in different domains, but none has yet been proposed in the domain of CV comprehension. Table 2.1 summarizes the approaches in related domains.

2.2 Word embedding

For decades, researchers have been searching for ways to represent words so it can be used in downstream models. However, the representation was mostly limited at the synthetic level, which only focus on arrangement of words in the sentences. In 2013, Tomas et al [25] proposed two models (Word2Vec) that can represent single words in vector space, providing both synthetic and semantic properties, which means it can also capture the meaning of words. Following this, another widely popular GloVe[26] was proposed. A study of word embedding impact on RC by Dhingra et al[27] showed that the use of pre-trained GloVe vectors can outperform

Table 2.1: Comparison of different approaches used in related domains

Related domains	Approaches used	Remarks
Machine Reading Comprehension	Name Entity Extraction	Need to maintain domain data dictionary
	Domain ontologies, taxonomies	Need to build or reuse relevant ontologies
	Extract information with semantic properties	Can only perform on manually pre-segmented CV sections.
Community Question Answering	Learning to rank	Require labelled questions and answers
	Similarity between Query and Existing question	Require a reliable vector representation

word embedding such as Word2vec, and even those trained on the target corpus. This observation could prove useful during the development of this work.

2.3 Sentence embedding

For tasks such as RC, it is insufficient to rely only on word embedding since the context of a document typically rely on sentences and paragraphs rather than individual words. To represent a variable length sentence with a fixed length vector, one way is to perform an averaging of the word vectors in the sentence[28]. However, the word order within the sentence would not be considered, potentially losing information. Le et al[29] proposed representing a paragraph by a dense vector which is trained to predict words in the paragraph. Palangi et al[30] proposed to use Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM) cells to produce a semantic sentence representation. Both architectures allowed the word order information to be retained. Despite that, Wieting et al[31] observed that the simpler word averaging models could perform well for sentence similarity and entailment tasks as compared to LSTM models.

Chapter 3

Data analysis and insights

This chapter introduces the data set made available by the Recruitment Agency, after which an analysis is performed and insights reported.

3.1 Data introduction

The Recruitment Agency has 2 active systems, the v7 system and the newer vX system. The vX system comprises a better organized and complete data set. This work will focus on the data made available from the vX system.

3.1.1 Data Nomenclature

The following describes the domain nomenclature used throughout the report. A *Client* refers to the recruiting corporation. An *Opportunity* refers to the advertised position by the client. An *Application* refers to an application form that is customized for every unique Client. A *Candidate* refers to the applicant to an Opportunity. A *CV* refers to the Curriculum Vitae of a Candidate. Any description of a term enclosed in square brackets [] is an indication of substitution.

3.1.2 Data Architecture

The vX systems data architecture is summarized in Figure 3.1. There are 6,805,751 Applications, 176,242 Opportunities and 3,609,305 CVs, which took up a total of 703GB. The large dataset meant that performance of any downstream models is important. The Applications and Opportunities data sets are placed in separate folders after which they share the same structure within their folders. A folder named

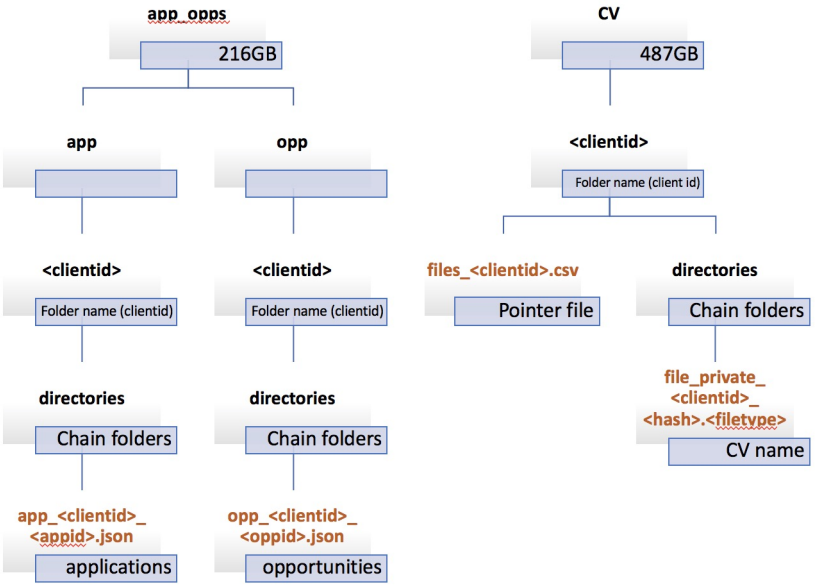


Figure 3.1: Architecture of data set

[clientID] first distinguish between the different Clients, under which a chain of folders sorts individual Applications and Opportunities as JSON files. A *[clientID]* folder first distinguish the CVs between the different Clients, under which, a chain of folders sorts individual CVs as raw files of varying file types. A single Comma Separated Value (CSV) file in each *[clientID]* folder attempts to keep track of the CVs and its parent Applications and Opportunities.

3.1.2.1 Clients

Each Client can be represented by a combination of the Applications they received, the Opportunities they offered, and the Candidates who applied for their Opportunities. Each client is essentially a different dataset, and are independent from each other. This means that an Application or Opportunity for client A cannot be directly correlated to one from client B. However, it is possible for identical CVs to be uploaded into two different client datasets. There are a total of 89 Clients; their industry distribution is depicted in Figure 3.2. For this work, one client from the financial industry was used.

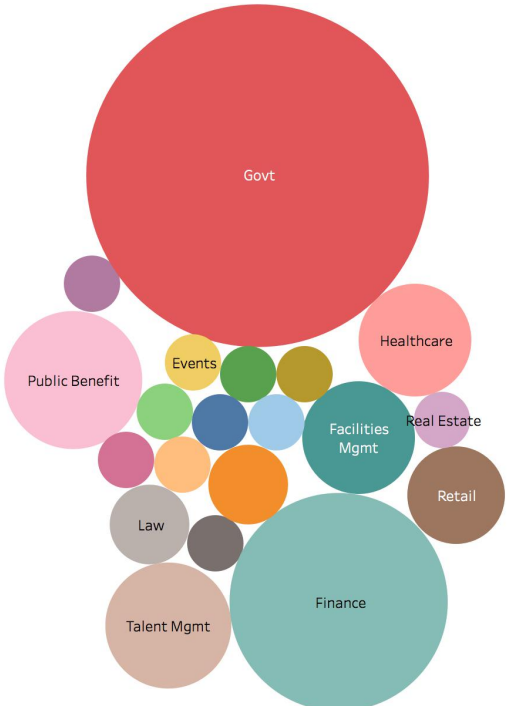


Figure 3.2: A distribution on the industries for the 89 clients in the dataset. The size of the circles is directly proportional to the number of clients. The government sector takes up the largest portion of the dataset, with the finance sector coming up second largest.

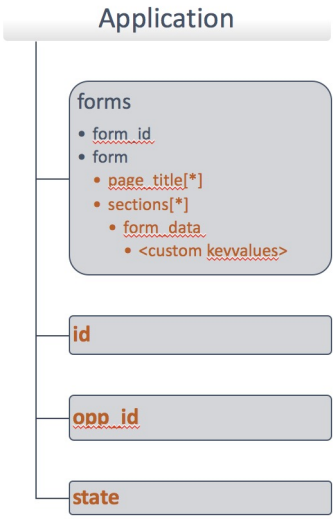


Figure 3.3: Typical structure of an Application. *pagetitle* represents broad questions such as 'Education' and 'Work Experience'. Typical key-value pairs reflect specific questions and answers corresponding to the broad question

3.1.2.2 Applications

A typical Application is represented by a web form, containing desired candidate attributes from each unique Client and represented by a *app_[clientid]_[appid].json* file name. The content of each file is described in Figure 3.3, with the notable fields highlighted in orange.

Within the file, *page title* refers to a broad question such as Personal Details and 'Work Experience', they are followed by multiple sections that can contain an unconstrained set of key-value pairs. Typical key-value pairs reflect a specific question and answer corresponding to the broad question. The key-value pairs vary from client to client. *Id* refers to the unique identifier of an Application for each unique client. *Opp_id* refers to the unique identifier of an Opportunity. *State* refers to the status of an Application for the Opportunity. *Statue* do not follow any fixed nomenclature, in fact, it could vary within Applications for the same Client. For example, if an application has been rejected, the status could reflect 'Rejected' or 'Not considered' or any other English sentence with similar meaning.

3.1.2.3 Opportunities

A typical Opportunity is represented by a web form containing a Client advertised position. Each Opportunity is represented by a *opp_[clientid]_[oppid].json* file. The content of each file is described in Figure 3.4, with the notable fields highlighted in orange.

In the file, *Title* refers to the Opportunity title. *Id* refers to the unique identifier of the Opportunity. *State* refers to the status of this Opportunity and do not follow any fixed nomenclature. Opportunities would not be discussed further for the intent of this work.

3.1.2.4 Curriculum Vitae (CV)

CVs are resumes uploaded as Application attachments by the Candidates. In general, there are two types of CVs, the graduate CVs and lateral CVs. Graduate CVs are collected during specialized graduate events while lateral CVs are collected on an ad-hoc basis. A single *file_[clientid].csv* for each Client tracks the CVs uploaded

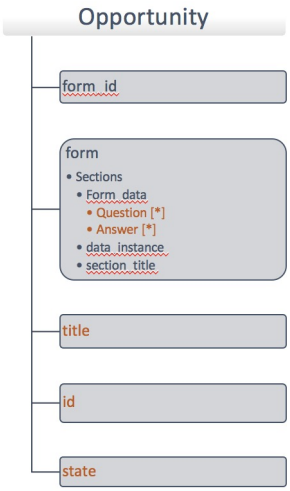


Figure 3.4: Typical structure of an Opportunity form

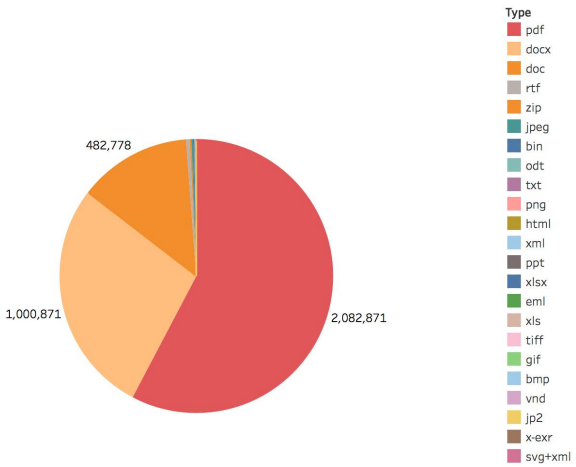


Figure 3.5: The distribution of file formats for the CVs in the dataset. The largest set of CVs comes in the form of PDF, DOCX and DOC formats.

by a Candidate, with a pointer to the relevant Application and Opportunity. A full path to the file is not available though, and it is not guaranteed that the CV can be found under the *[client_id]* folder. There were no restrictions on the type, format and content of the CVs being uploaded. Figure 3.5 shows the distribution of file types across all Clients. The most common type of CVs are PDF, DOCX and DOC formats.

3.2 Insights

3.2.1 Semantic Similarity

The broad question and key-value pairs (specific questions and answers) in Application forms varies from client to client but they could semantically mean the same thing. Each Application's state is labelled in an ambiguous but semantically similar manner, which means there are no reliable status label for each Application. There is significant ambiguity in the representation of states, broad questions, and answers for each application. This implies that the traditional categorization of data may be tedious and not achievable within a short frame of time. Furthermore, any downstream applications would suffer due to the need to update these categories. As such, an approach based on semantic similarity might be useful in the long run.

3.2.2 Data retrieval

The way that the current CVs are stored means it would require significant pre-processing before any meaningful work can be performed. It should be of good sense that the current structure not be inherited in follow up development of this work.

3.2.3 Language Disparity

It was implied that all CVs uploaded was in the language corresponding to the Applications. However, an analysis found that not all CVs were uploaded in the English language. For this work, content in foreign languages ought to be filtered.

3.2.4 CV Splitting

The Recruitment Agency do not possess the technology to verify individual CVs, as such, it is not guaranteed that each uploaded attachment does contain a valid CV. There is a need to filter irrelevant attachments during pre-processing. Due to the distribution of CV types and the unconstrained formats used in the CVs, it is extremely difficult to reliably extract sections of a CV. Other than utilizing in-depth ontologies, it could be possible to ride on recent advancement of semantic word embedding techniques in understanding the CVs.

Chapter 4

Proposed Approaches

This chapter will first formulate the problem of answering Applications using CVs uploaded in the recruitment process. Three different approaches that address the problem will then be discussed.

4.1 Problem formulation

The goal is to answer the broad questions of an Application with information from a CV. Mathematically, the problem can be formulated as follows. Let the broad question of the Application be denoted as Q. Each answer A_{qa} to the broad question Q is represented by a concatenation of specific questions q and their respective answers a from the Application. Each q and a are represented by a set of words w_i, where i represents a variable ith word in the set. This is represented in Equations 4.1 through 4.4

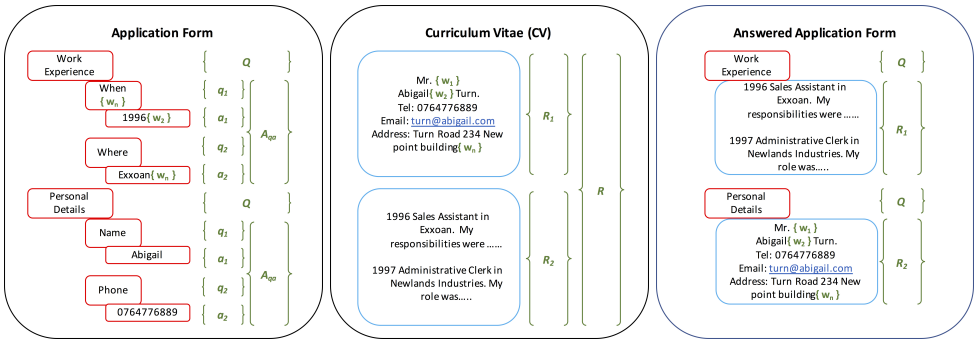
$$Q \leftarrow A_{qa} \tag{4.1}$$

$$a = [w_1, w_2 ... w_i] \tag{4.2}$$

$$q = [w_1, w_2 ... w_i] \tag{4.3}$$

$$A_{qa} = ((q,a)_1, ((q,a)_2 ... ((q,a)_n) \tag{4.4}$$

Let a CV be denoted as R. A CV can be broken down into variable sized sections denoted by R_n, where n represent the nth section of the CV. Each section R_n is represented by a set of words w_i, where i represents a variable ith word in the set. Given



(a) Application form **(b)** CV **(c)** Answered form

Figure 4.1: A realistic representation of the problem formulation. (a) Applications have broad questions that needs to be answered. (b) CVs are broken down into sections. (c) Sections of an applicant's CV will be used to answer the broad question. Content of Applications and CVs are largely different in terms of format and language structures, giving this task an added challenge.

an unseen CV section R_n, the corresponding broad question Q is to be determined. This is represented in Equations 4.5 through 4.7

$$R = [R_1, R_2 ... R_n] \tag{4.5}$$

$$R_n = [w_1, w_2 ... w_i] \tag{4.6}$$

$$R_n \rightarrow Q_? \tag{4.7}$$

For this work, the broad questions are defined as 'Education', 'Skills', 'Personal Details' and 'Work Experience'. Figure 4.1 shows a visual representation of the problem formulation alongside a realistic example. The notable challenge in this problem is that the content of Application and CVs are largely different in terms of format and language structures. The candidate filled Application are highly structured, static in format and contain bit and pieces of information pertaining to specific questions and answers. The CVs on the other hand are unstructured and vary in presentation formats. They are however written in well-defined languages (E.g. English).

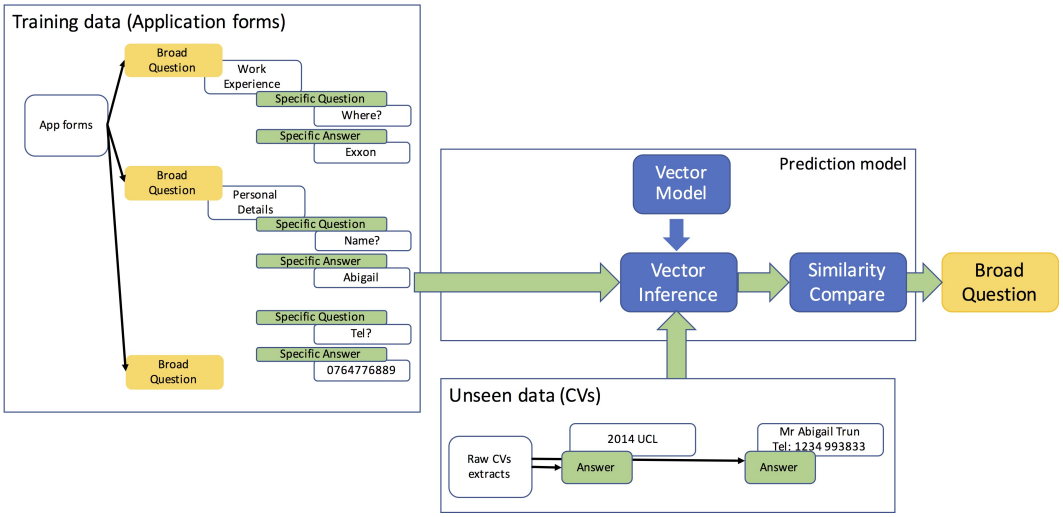


Figure 4.2: Overview of Inverse Answering by Similarity approach. The vector representation model would be built using a training set of CV Sections (Not shown in figure). The prediction model would compare the similarity between the items in the Applications and the unseen CV section before arriving at the most likely broad question for the CV section.

4.2 Inverse Answering by Similarity (SIM)

The motivation for this approach is that the type of answers entered for each broad question Q should be similar across candidates, and these answers would typically be available in uploaded CVs, albeit in structures very different to the receiving Application. For example, when an Application asks for personal details in a structured form, similar information would typically already be available in a CV (E.g. email address). The Inverse Answering by Similarity approach attempt to determine the similarity between the content of Applications under each broad question Q, and the content of CV sections R_n. Figure 4.2 shows the overview of this approach. First, a vector representation that would represent A_{qa} and R_n in the same space must be defined. Each vector representation would be computed from a Vector Representation Model. This model is represented in Equation 4.8.

$$V = [W_1, W_2 ...., W_i] \tag{4.8}$$

where V represent the vocabulary of the vector representation model, W_i represent a word in the vocabulary. From V, vectors of fixed sizes for each broad question

Q would be created with the content from the corresponding A_{qa}. In the same manner, vectors for each CV sections R_n would be created. The created vectors are represented in Equations 4.9 and 4.10

$$D_{A_{qa}} = [n_1, n_2, ...n_n] \tag{4.9}$$

$$D_{R_n} = [n_1, n_2, ...n_n] \tag{4.10}$$

where $D_{A_{qa}}$ and D_{R_n} represent the vectors for the answer to the broad question Q and the vector of the CV section R_n respectively. n_n represent a number element in a vector of n dimensions.

Given an unseen CV section R_n, a similarity score would be computed against every A_{qa} via their computed vectors. The similarity scores are then sorted and the M most similar A_{qa} would be selected. Note that by definition, each answer A_{qa} would correspond to a broad question Q as shown in Equation 4.1. As such, a probability of each Q that corresponds to the unseen CV section R_n can be computed in Equation 4.11. where $C(x)$ is the number of elements of $1, 2...M$ that are equal to x, where x is one of the classes of Q.

$$P(Q_x) = \frac{C(x)}{M} \tag{4.11}$$

The prediction will be based on the highest probability as follows.

$$Q = \text{argmax}_{j \in [|Q_1|, |Q_2|, ..., |Q_n|]} \tag{4.12}$$

Discussion on the vector representation model and the similarity scoring function are presented in Chapter 4.5

4.3 Inverse Answering by Classification

With the availability of computing power and ever-growing data sizes, a Machine Learning approach that rely more on learning from the evolving data can be taken in solving the problem. In the most basic form, the Inverse Answering by Classifi-

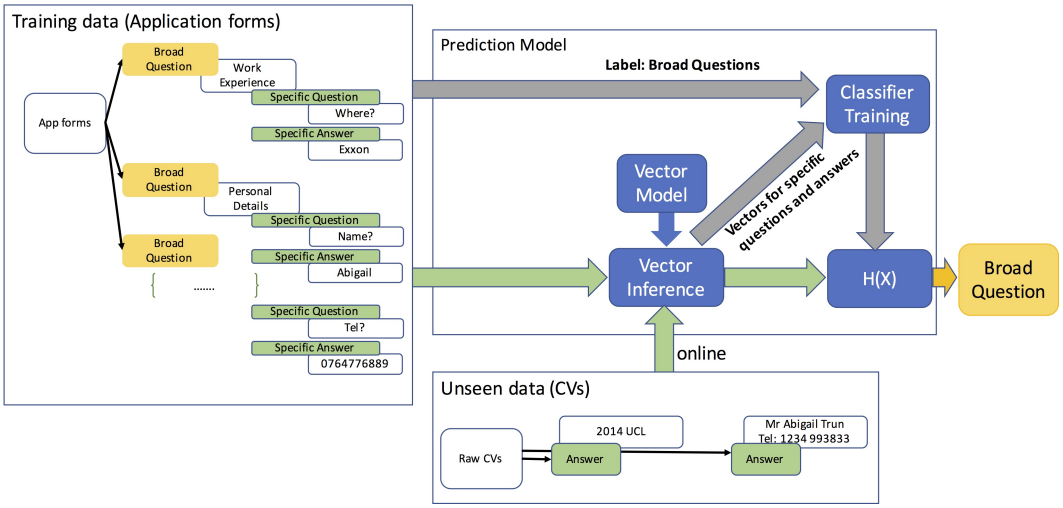


Figure 4.3: Overview of Inverse Answering by Classification approach. The vector representation model would be built using the training set of CV sections (Not shown in figure). The prediction model would be trained using the specific questions and answers from Application. It would then predict the most likely broad question for the CV section.

cation approach takes the vector $D_{A_{qa}}$ as defined in Equation 4.9 as training features and broad question Q as corresponding training labels to train a classifier. Given an unseen CV section (R_n), the vector D_{R_n} is generated by the Vector Representation Model and then the trained classifier (prediction model) is used to predict the label Q. Figure 4.3 shows the overview of this approach. The classes of the classifier would be represented by the respective broad question Q as described in Equation 4.13.

$$C = Q_1, Q_2 ... Q_p \tag{4.13}$$

where p is the total number of unique broad questions in the Application. For training, an input feature vector X_t is represented by Equation 4.14 where X_t is the vector of the answer A_{qa}.

$$X_t = D_{A_{qa}} \tag{4.14}$$

The feature vector that will be used for prediction is represented by Equation 4.15, where X_e is the vector of the unseen CV section R_n.

$$X_e = D_{R_n} \tag{4.15}$$

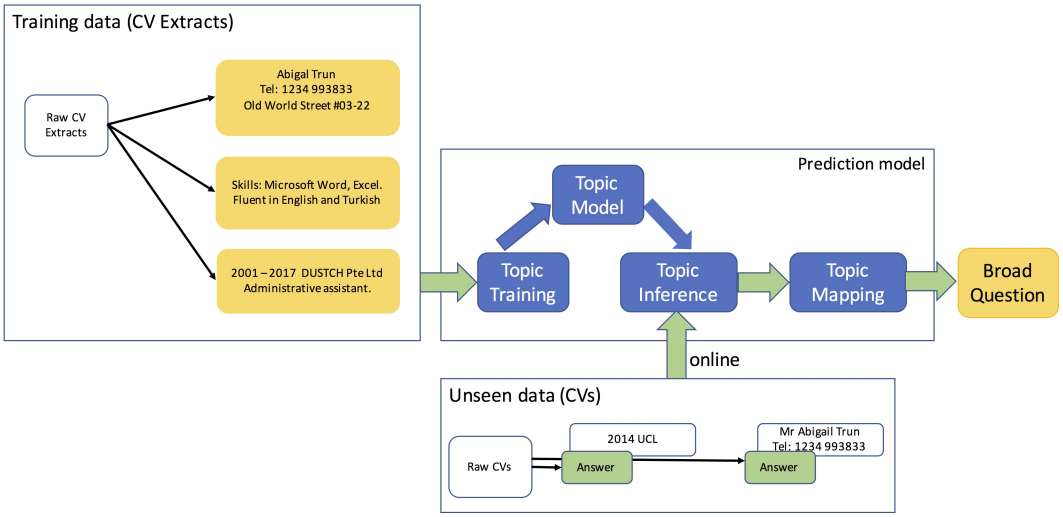


Figure 4.4: Overview of the Inverse Answering by Topic Modelling approach. A topic model of the entire training set of CV sections is first built. Topics are manually mapped to corresponding broad questions. The topic-word distribution of each unseen CV can now be inferred and thus a prediction be made on the CV section's broad question.

Different types of machine learning classifiers performed differently in different domains, as such it is necessary to experiment with different algorithms before determining the best algorithm for the Inverse Answering by Classification approach. The algorithms to be used in the experiments are elaborated in Chapter 5.6.2.

4.4 Inverse Answering by Topic Modelling (TOPIC)

The motivation of this approach is that the words in a CV section is likely a result of a typical broad question answered in a CV. For example, a CV section answering education history tend to contain more words such as 'University' and 'Education', than words such as 'Experience'. Similarly, a section detailing work history tend to contain words such as 'Experience' and 'Work', rather than words such as 'Hobbies'. Generalizing, it would make theoretical sense that each document would likely to have a high distribution of words from a topic. Figure 4.4 shows the overview of this approach.

The approach first investigate if the topic distribution of the entire CV section corpus can be intuitively assigned to the classes of the broad question Q. If the previous step turns out true, it would mean that it is possible to 'classify' an unseen

CV section to an appropriate broad question Q simply by analyzing its topic distribution. For example, a CV section may have a topic distribution of 0.7,0.1,0.1 and 0.1 for four topics. The topic with the highest probability would be selected as the predicted topic. This is in turn mapped to the human assigned broad question in the investigation phase of this approach.

4.5 Vector Representation Models

A challenge at hand was that there are two distinct text structures. In a highly structured Applications, the content is extremely short and hardly form any context. In CVs, the content is unstructured but they are typically full of context. For the reasons above, vector representation models known for typical text classification tasks may not perform as expected. As such, it was necessary to investigate the vector representation model that can perform in this problem. The following describes the vector representation models that will be investigated to create the vectors $D_{A_{qa}}$ and D_{R_n}.

4.5.1 Term Frequency (TF)

The TF model represent a text document by the occurrences of the words (term frequency) in the document. For example, given a toy corpus of *'This is the weather from twenty three twenty three hours'*. A Vector Representation Model can be built with the vocabulary;

$$V_{BOW} = [This, is, the, weather, from, twenty, three, hours] \tag{4.16}$$

Then for a toy document with the content *'This is the twenty fifth times in the frame of twenty hours'*, it can be represented as;

$$D_{BOW} = [1,1,1,0,0,2,0,1] \tag{4.17}$$

This model is also known as the Unigram model, where each word is counted on its own. For this work, the experiments focused unigram models.

4.5.2 Term Frequency Inverse Document Frequency (TFIDF)

TFIDF prescribes a heavier weight to a word if it has high occurrences in a document but low occurrences in the document collection. The assumption is that a word that occurs in all the documents in the corpus has no discriminative power. The Term Frequency (TF) is as what is described in Section 4.5.1. The Inverse Document Frequency (IDF) determine how many documents in the collection does the word appear (DF) and then the inverse is taken as depicted in Equation 4.18

$$IDF = log\frac{N}{DF} \tag{4.18}$$

$$D_{IDF} = \begin{bmatrix} idf_{this} \\ idf_{is} \\ idf_{the} \\ idf_{weather} \\ idf_{from} \\ idf_{twenty} \\ idf_{three} \\ idf_{hours} \end{bmatrix} \tag{4.19}$$

The product of term frequency and inverse document frequency is taken to produce TFIDF.

$$D_{TFIDF} = D_{BOW} x D_{IDF} \tag{4.20}$$

4.5.3 Word2vec(W2V)

A W2V model represent words in a fixed size vector.

$$W = [n_1, n_2, ...n_n] \tag{4.21}$$

where W represent a word and n_n represent a number element in a vector of a fixed n sized dimension. This only represent each word by a vector, however, it would be necessary to represent a string of words by this fixed sized vector. To solve this, for a document of N words, the word vectors corresponding to the words in the

document are averaged to form a document vector as depicted in Equation 4.22. Out-Of-Vocabulary words are ignored and will not count for the averaging.

$$D_{W2V} = \frac{1}{N}\sum_{i=1}^{N}[n_1, n_2, ...n_n]_i \tag{4.22}$$

The advantage of W2V over TF and TFIDF is that each vector can be represented in a fixed sized vector, typically multiple magnitudes lesser than that of TF and TFIDF. This means that the speed of processing using W2C would be significantly faster than that of TF and TFIDF. Another advantage of W2V is that the information contained within the vector retains a semantic meaning to the word. In other words, vectors for semantically similar words such as 'chair' and 'sofa' can also be mathematically similar.

4.5.4 Paragraph2Vec (D2V)

Paragraph2vec[29] is a document vector representation of sentences and documents like W2V. The difference is that each vector of fixed n dimensions already represent a document in its entirety. D2V vectors contain information that address the order of the words in a document.

$$D_{D2V} = [n_1, n_2, ...n_n] \tag{4.23}$$

4.6 Similarity Measure

Cosine similarity was selected for the similarity computation between vectors $D_{A_{qa}}$ and D_{R_n}. This measure is selected firstly because of its interpretability and secondly for its wide use in several domains. The measure is represented in Equation 4.24

$$cosine\_sim = \frac{\sum_{i=1}^{n} A_{qa\_i} R_{n\_i}}{\sqrt{\sum_{i=1}^{n} {A_{qa\_i}}^2}\sqrt{\sum_{i=1}^{n} {R_{n\_i}}^2}} \tag{4.24}$$

where n is the size of the vectors and A_{qa_i} and R_{n_i} are the individual components of vectors $D_{A_{qa}}$ and D_{R_n} as described in previous sections.

Chapter 5

Experiments

This chapter highlights the objectives of the experiments before proceeding to describe the data, performance metrics and evaluation approaches used in the experiments. Next it discuss the experiments and results for the vector representation models and prediction models, Finally, a discussion is made on the effects of data preprocessing.

5.1 Objectives

The objectives summarized in Figure 5.1 set the course of experiments with the goal of systematically assessing the three approaches defined in Chapter 4.

5.1.1 Compare vector representation models

Completion of this objective answers the following questions. Will state of the art D2V perform better? Are the models suitable for use on practical applications?

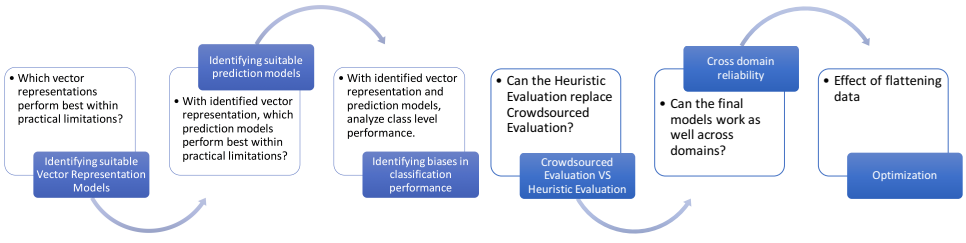


Figure 5.1: Objectives are to identify suitable vector representation and prediction models,then to answer questions such as classification performances, evaluation approaches and cross domain reliability. Last objective is to evaluate effects of data preprocessing on the proposed approaches.

Which vector model should be selected to represent the data at hand?

5.1.2 Compare proposed approaches

The next objective investigates the effectiveness of proposed approaches with the vector representation model of choice. This answers questions on their F1 performance scores and practicality in real world scenarios.

5.1.3 Classification performance

The next objective investigates if some Application questions are answered more accurately than others. This helps in future refinement of the approaches for a more balanced outcome.

5.1.4 Assess evaluation approaches

The next objective investigates the robustness of the Heuristic Evaluation described later in Chapter 5.4.2. If the Heuristic Evaluation shows promising results, it can solve the Recruitment Agency's problem of expensive data labeling on the CVs.

5.1.5 Effectiveness across domains

The experiments focuses on CVs from graduates applying in the financial domain. The best performing approaches are tested on two unseen sets of CVs from different domains to assess their effectiveness across domains.

5.1.6 Optimization

5.1.6.1 Effects of data pre-processing

The final objective investigates the effects of preprocessing the raw data before applying vector conversion.

5.2 Setting up the data

This section described the data used in the experiments. A set of Applications with attached CVs were made available. Specifically, Applications and respective CVs from the financial domain were used. The Inverse Answering by Similarity approach uses the Application records ((A_{qa}) defined below as the training set for similarity comparison against unseen CV sections. The Inverse Answering by

Classification approach uses the same Application records and their corresponding broad questions as the input features and labels respectively. The Inverse Answering by Topic Modelling approach uses the split CVs (CV sections defined below) as training for the topic model.

5.2.1 Application forms

A total of 380,688 Applications from financial domain was made available. Figure 4.1a shows an example of how the text were engineered from the structured application forms. Each Application holds a set of broad questions, to which specific questions are answered by candidates. The specific questions and answers were concatenated together to form a whole answer to the broad question. For example, a specific question 'What is your first degree' and its specific answer 'UCL', are concatenated to form a single answer (A_{qa}) to the broad question 'Education'. The reason for including the specific question was to provide a textual context to each specific answer, which was often too short to provide enough context. To ensure a good quality of data, Applications that do not contain specific answers were omitted. These were then re-organized to fit into 4 broad categories namely, Education, Skills, Personal Details, and Work Experience. For example, original broad questions such as 'Basic Education', 'Pre-University Education, Current Education' were all organized into a single category 'Education'. As a result, a total of 1.3 million records (A_{qa}) were created, each uniquely labelled to 'Education', 'Skills', 'Personal Details' or 'Work Experience' respectively. These records are used as training data for the Inverse Answering by Similarity and Inverse Answering by Classification approaches.

5.2.2 CVs

5.2.2.1 Converting raw CVs to text

CVs must be converted from their native formats into a machine-readable text based format. Utilizing open-source tools PDFTOTEXT[32] and DOCX2TXT[33], a conversion script was implemented to convert all PDF and DOCX CV formats to text formats. The two file formats were chosen because of their significant volume in the

Recruitment Agency's database. A language detection module was implemented to detect and filter all Non-English CVs. The conversion software suffered from encoding errors in some parts of the CVs, thus a screening module was implemented to remove non UTF-8 encoded CVs to avoid problems downstream. As a result, a remaining 289,466 out of 316,825 CVs were available for downstream processing.

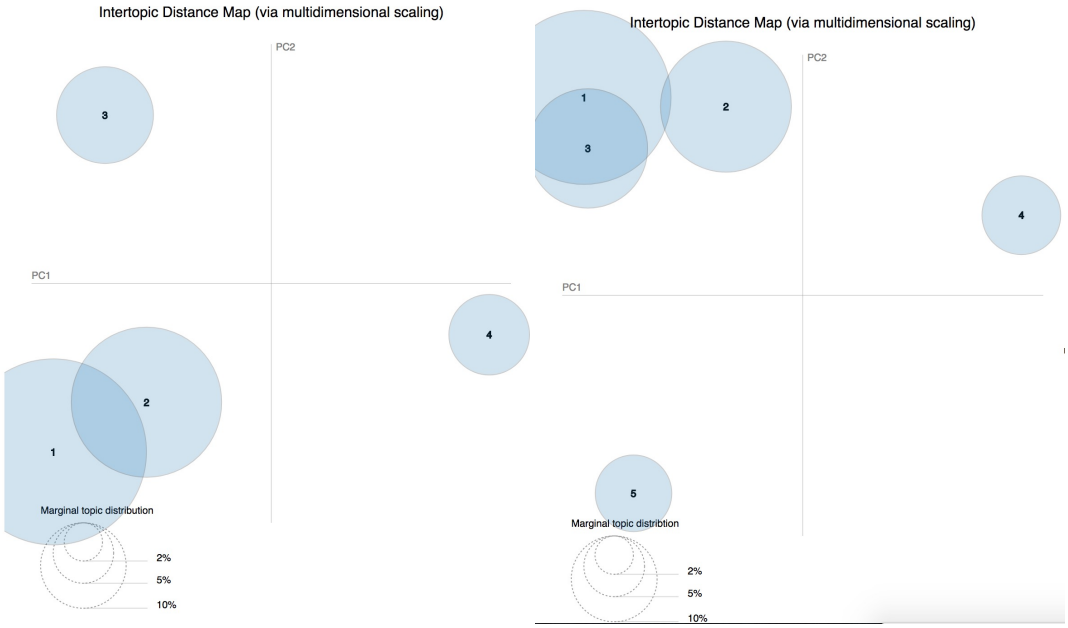
5.2.2.2 Splitting CV into sections

For a broad question in an application form, the answers are typically found only in certain sections of a typical CV. For example, you would typically need to read a small part of a CV to get the personal details of the applicant. Therefore, it was necessary to break the CVs into smaller sections. A simple approach was to screen the CVs for sections separated by multiple carriage returns, and for sections separated by isolated capitalized words. A software module was implemented to perform this operation. As a result, 2,044,032 CV sections were made available for downstream processing.

5.2.2.3 Validate CV sections

How to ensure that the split CV sections made sense? Typical CVs contain four types of topics namely, 'Personal Details', 'Education, Work Experience' and 'Skills'. If there's a way to find a clear distinction of these topic in the set of CV sections, it would validate the splitting of the CV sections. Likewise, if the CV sections were badly done, the splitting algorithm needs to be refined.

A popular topic modeling algorithm Latent Dirchlet Allocation(LDA)[34] was employed to discern if the typical CV topics could be discovered. First. about 40 percent of all CV sections were filtered and pre-processed to convert the content into lower case, and to remove stop words and punctuation. LDA took the processed CV sections and represented it as a DxW section-word matrix, where D represent the total number of CV sections, and W represent the total number of unique words in the entire CV section corpus. Matrix factorization was then performed on the matrix. In real number factorization, it meant decomposing the number into more numbers such that the product of these numbers equaled the original number. In matrix factorization, the same intuition applies. For example, this matrix of shape



(a) LDA results with k=4 **(b)** LDA results with k=5

Figure 5.2: Results of the LDA model applied onto the test set of CV sections. The results showed that the documents can be be broken down into four topics and demonstrates the validity of the sectioning of the CVs.

DxW could be decomposed into a DxK and a KxW matrix where $K << D$ and $K << W$. The resulting matrices would represent the topic-document and topic-word distribution respectively. In this case, if we set $K = 4$, it is analogous to finding four topics. LDA further improved the distribution in the matrices with sampling techniques.

Although the goal was to discover four topics, this work performed an LDA analysis on both four and five topics. The results were analyzed with LDAVis[35]. Figure 5.2 shows the topics plotted as circles with the distance between circle centers analogous to the distance between topics. in Figure 5.2b, there are five topics, with topics 1 and 3 being a single topic since they are mostly overlapped. A separate analysis was performed with $K = 4$, which shows a similar conclusion of four distinct topic groups.

An analysis of the 20 top most relevant words of each topic shown in Table 5.1 showed a strong resemblance to words that would appear in 'Education', 'Skills', 'Personal Details' and 'Work Experience' respectively. This showed that the algo-

Table 5.1: Analyzing the top 20 most relevant words for each topic in the generated topic model by LDA. By observation, topics 1,2,3,4 were identified as 'Work Experience','School Extra Curricular Activities','Education','Skills' and 'Personal Details' respectively.

Topic 1	Topic 2	Topic 3	Topic 4
financial	university	skills	com
research	school	english	edu
management	finance	interests	gmail
investment	business	languages	edu
2014	economics	excel	email
analysis	education	microsoft	address
intern	2013	fluent	1
2013	3	office	0
business	2012	c	uk
new	2014	computer	mail
experience	management	language	street
market	gpa	french	mobile
company	science	basic	2
data	2011	word	road
2012	mathematics	native	ny
project	4	powerpoint	phone
clients	financial	spanish	44
team	2010	ms	street
summer	bachelor	proficient	london
sales	college	bloomberg	6

rithm to break the CVs into sections was sufficient for this work.

5.3 Preprocessing of data

All content were converted to lower case. All stop-words and punctuation were also removed. Contiguous alphanumeric characters adjacent to non alphabets and numbers were also separated into individual words. For example, myMAIL@mail.com was split into 'mymail', 'mail' and 'com'.

5.4 Setting up for evaluation

5.4.1 Crowd-sourced evaluation

As CV sections were automatically generated from whole CVs, it was not known which broad question they answered to. To evaluate the above approaches, it was

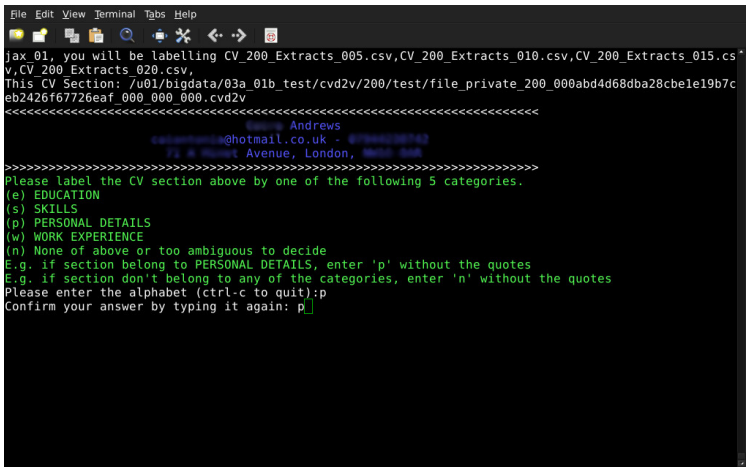


Figure 5.3: A screenshot of the implemented labelling software with some data masked to protect applicant's privacy. Each CV section was displayed in blue while the main instructions were displayed in green. After confirming the answer, the next CV section would be displayed to the user for labelling.

necessary to get humans to label the CV sections according to the broad questions they might answer to. A total of 80 unseen CVs from the financial industry were labelled by 5 volunteers, with 10 of the CVs repeatedly labeled by all volunteers. Each volunteer labeled an average of 10 sections per CV, with each section corresponding to answers for 'Education', 'Skills', 'Personal Details', 'Work Experience' and 'None of the above'. A total of 250 valid CV sections were successfully labelled. A screen shot of the implemented labelling software is shown in Figure 5.3. To assess the quality of the labelled CV sections, a measure of reliability of agreement across the volunteers was performed. The Fleiss's Kappa (k) score was used on the identical set of CVs that was labelled repeatedly by each volunteer to measure the agreement level across all labelers as depicted in Equation 5.1.

$$k = \frac{\bar{P} - \bar{P}_e}{1 - \bar{P}_e} \tag{5.1}$$

where $\bar{P} - \bar{P}_e$ refers to the degree of agreement that is attainable above chance, and $1 - \bar{P}_e$ is degree of agreement achieved above chance. With 5 classes and about 100 CV sections labeled repeatedly by all labelers, a 100x5 agreement table was created. Let i represent the CV section, j represent the class and n_{ij} represent the number of labelers who assigned the ith CV section to the jth class. To compute $\bar{P}$

and $\bar{P}_e$, first compute P_i and P_j as follows.

$$P_i = \frac{1}{n(n-1)} \sum_{j-1}^{k} n_{ij}(n_{ij}-1) \tag{5.2}$$

$$P_j = \frac{1}{Nn} \sum_{i=1}^{N} n_{ij} \tag{5.3}$$

where

$$1 = \sum_{j-1}^{k} P_j \tag{5.4}$$

where N is the number of CV sections and k is the number of classes. Thus $\bar{P}$ and $\bar{P}_e$ were computed as follows.

$$\bar{P} = \frac{1}{N} \sum_{i=1}^{N} P_i \tag{5.5}$$

$$\bar{P}_e = \sum_{j=1}^{k} P_j^2 \tag{5.6}$$

A Fleiss's Kappa score of 0.77 was achieved which put it at substantial agreement across all labelers[36]. This level of agreement was acceptable and thus no further adjustments was needed for the labelling process.

After removing sections marked invalid by users, and balancing the classes within the valid CV sections, a total of 144 CV sections were available as unseen test data. The evaluation took the labeled CV sections and used them to evaluate all the three proposed approaches described in Chapter 4.

5.4.2 Heuristic evaluation

Although the correctness of Crowdsourced Evaluation is guaranteed, it would require significant human resources to label and the evaluation sample size would most likely be limited. To overcome this problem, a Heuristic Evaluation was proposed. The idea was that since any CV would be accompanied by an Application, the Application's content could be used to determine if the prediction of the broad question of the CV section was correct. When a broad question had been predicted to a CV section, the specific questions, and answers of the broad question under the

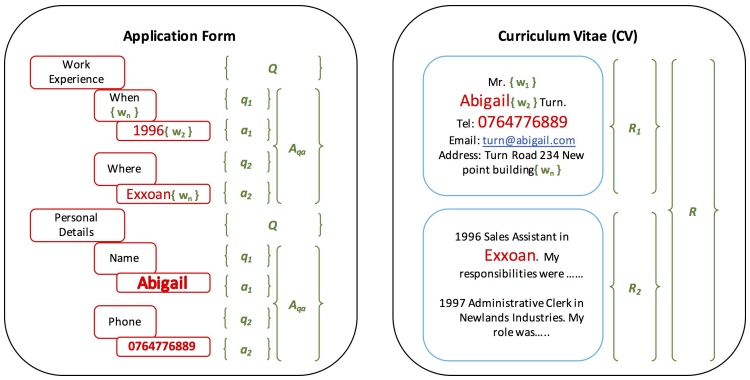


Figure 5.4: Understanding Heuristic Evaluation. Every CV comes with an Application. This means we predicted a CV section to answer to 'Personal Details', we should find exact words in the corresponding answers of the broad question in its Application. If a certain number of non-stopwords such as names (highlighted in red) appears in both Application answers and CV section, take it to be a HIT, else it would be a MISS.

Application would be compared with the content of the CV section. If the prediction was correct, there should be at least one (or multiple) non-common words that are identical and the outcome would be designated as a 'HIT'. if there are no identical words, it would be designated a 'MISS'. The number of the identical words wa designated as M. In the experiment, M was set at 1 for 'Personal Details', and 3 for 'Education', 'Skills' and 'Work Experience'. Figure 5.4 provides a visual understanding of this approach.

5.5 Selecting performance metrics

Different performance metrics cater to different kind of problems and data distribution. In this experiment, a common metric must be used to evaluate the performance of the 3 approaches. The following sections justified the choice of the performance metrics used, with the technical computation detailed in Appendix A.

5.5.1 Accuracy

In this work, accuracy was exclusively used for Heuristic Evaluation due to its HIT/MISS methodology. When comparing Heuristic Evaluation with Crowdsourced evaluation, accuracy would be used as well. Technical details of the accuracy metric is presented in Appendix A.1.

5.5.2 Multi-Class F1

Although accuracy is a simple metric and works well in a balanced test set for overall classification performance, it doesn't help in analyzing how well the approaches perform in classifying the different classes. Secondly, in an uncontrolled practical setting, the test data is expected to be unbalanced, with excessive tilt towards the 'Work Experience' class. With the above considerations, the F1 score, which is the harmonic mean of precision and recall metrics was exclusively used in the Crowdsourced Evaluation. Details on computing the multiclass F1 metrics is presented in Appendix A.3.

5.5.3 Significance Testing of results

In some situations, a significance test was performed to determine if the performance difference between the approaches was by chance. The selection of a significance test was dependent on the following assumptions.

- All approaches are tested on matched samples.
- The evaluation is performed on a small labelled set.
- The distribution of the population cannot assumed to be normal.

As a normal distribution cannot be assumed, only non-parametric tests were considered.

5.5.3.1 Generalized McNemar's Test

McNemar's Test[37] work on pairwise agreement to determine if the performance difference was significant. This test could however only be performed on binary classifiers thus a generalized version of the McNemar's test was required. This generalized version is also known as the Stuart-Maxwell Test[38, 39]. The computation of the Stuart-Maxwell's Test is detailed in Appendix A.4.

5.5.3.2 Wilcoxon's Signed Test

The Wilcoxon's Signed Test[40] can be used if the following assumption were satisfied; Dependent samples, samples independently and randomly drawn, continuous and ordinal samples. By the nature of the experiment, the above assumptions are satisfied. To perform this test, it was necessary to evaluate based on multiple folds

of the labelled test set, and to retrieve a set of F1 scores. For example, instead of a single F1 score from each approach, the labeled test set is broken down into multiple test sets and then a F1 score was computed for each set. Two sets of F1 scores from two different approaches were then subjected to the Wilcoxon Signed Test.

5.5.3.3 Interpreting significance

For both significance tests, we set the null hypothesis H_0 that both approaches being compared are statistically no different, and an alternate hypothesis that they are different. The significance level is set at α=0.05. We either reject the null hypothesis at p-value$<=\alpha$ and accept alternative hypothesis that one approach is significantly better than the other, or accept the null hypothesis that the pair of approaches were statistically not different on the data sets at p-value$>\alpha$.

The Wilcoxon's Signed Test did not work well when samples are limited, while the Stuart-Maxwell Test cannot be computed if there's a case where the sum of row or column in the agreement table was zero. As such, the Stuart-Maxwell Test was the primary significance test, if it fails, the Wilcoxon's test was used.

5.6 Results

5.6.1 Comparison of Vector Representation Model

Table 5.2 shows the properties of various vector representation models. The dimension size (dim) referred to the fixed number of elements in the vector representation as discussed in Chapter 4.5. To get the impact of how this translate to the sizes of the features, the features size of each vector representation on a set of 150,000 Application sections were indicated in the Feature Size column. Note that this size also consisted of overheads meta stored in the features. With more than 2GB of space, the TF and TFIDF vector representation had higher storage requirements than W2V and D2V. Sparsity measures how sparse the feature vector was in the different vector representations. Both TF and TFIDF had sparsity of 0.99, which meant 99% of the 750M elements was zero valued. W2V and D2V had very dense vectors of sparsity zero. All the vector representation models had similar loading times, with a one-time model loading of 1 second for TF and 2 seconds for the rest of the models.

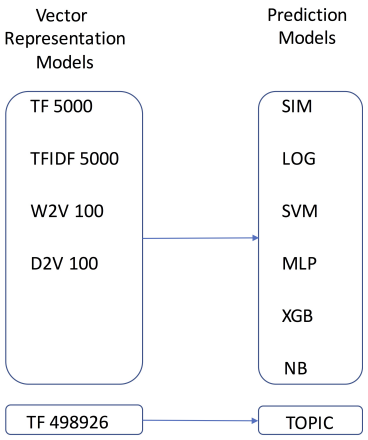


Figure 5.5: Pairwise experiments between various vector representations of predefined vector dimensions with various prediction models. The Inverse Answering by Classification approach experimented with Logistic Regression(LOG), Support Vector Machine (SVM), Multi-Layer Perceptron(MLP), Boosted Tree (XGB) and Naive Bayes (NB) algorithms. Topic modelling is the only model that will use the full vocabulary of the training corpus.

Each model had an inference time of less than a second, this was the time needed to infer a vector for 100 unseen documents.

Next, the effectiveness of each vector model's ability to discern similar CV sentences was evaluated. Table 5.3 shows that W2V was able to discern the similarity between two similar sentences. D2V perform below expectations with a score not much better than TF and TFIDF.

Finally, the effectiveness of each vector representation model was evaluated by pairing it with a downstream prediction models as depicted in Figure 5.5.

Figure 5.6 shows the F1 score of all the Inverse Answering by Similarity and Inverse Answering by Classification approaches under the TF 5000 Dim, TFIDF 5000 Dim, W2V 100 Dim and D2V 100 Dim vector representation models. An interesting observation is that all the approaches performed similarly to the dummy classifier (classifying at random) when they were using the D2V representation model. Except for the Naive Bayes (NB) classifier, all other classifiers consistently scored the highest when used with W2V vector representation.

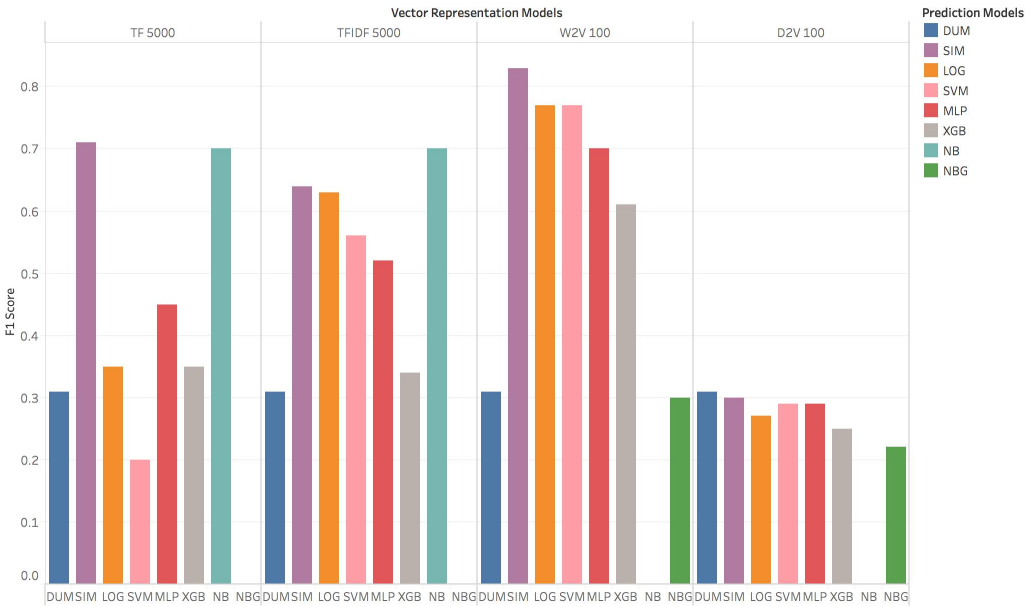


Figure 5.6: W2V vector representation model performed better across all approaches. D2V was consistently poor across all models. Its performance on various classifiers was compared to that of a dummy classifier which performs classification at random, the results suggested that D2V cannot represent the datasets in the experiments. The Inverse Answering by Similarity approach performed consistently better with all vector representation models compared to other approaches

Table 5.2: Dimension (dim) size refers to the size of each feature vector, this contributes directly to the feature size of 150,000 samples. Sparsity of 0 means there are only non-zero elements in vector samples. Model time shows time needed to load representation model for inference. Inference time refers to the time taken to infer 100 test documents

Vector Representation	Feature size(GB)	Sparsity	Model time(sec)	Inference time(sec)
TF 5000 dim	2.20	0.99	1.0	<1
TFIDF 5000 dim	3.70	0.99	2.0	<1
W2V 100 dim	0.37	0.0	2.0	<1
D2V 100 dim	0.37	0.0	2.0	<1

Table 5.3: An example of how well each vector model can represent 2 similar CV sentences that were both labeled as 'Education'. The sentences were converted to lower case, stripped of stop words and punctuation before being represented by each vector model. A cosine similarity score between the sentences were computed for each vector model. W2V performed much better than the baselines TF and TFIDF. Surprisingly, D2V performed very badly as well.

Preprocessed sentences	TF	TFIDF	W2V	D2V
page 1 3 education training institution providing education university st gallen switzerland start date end date september 17th 2012 february 16th 2015 thesis submission graduation autumn 2015 aspired title qualification master arts hsg m hsg banking finance mbf unisg ch average grade received 5 36 grading scale 1 0 6 0 1 0 worst 6 0 best pass grade 4 0 principal subjects financial markets financial institutions quantitative methods core subjects focus derivatives core elective courses ethics contextual elective courses additionally 40 ects supplementary credit	0.383	0.378	0.805	0.383
institution providing education university bayreuth germany start date end date september 1st 2008 march 31st 2012 title qualification awarded bachelor arts philosophy economics pe uni bayreuth de average grade received 1 5 grading scale 1 0 5 0 1 0 best 5 0 worst pass grade 4 0 exchange term location shanghai china exchange term institution shanghai international studies university china shisu edu cn exchange term start date end date august 31st 2010 february 16th 2011 principal subjects bayreuth lectures economics business administration philosophy shanghai full time mandarin course chinese language program achievements awards distinctions nomination best bachelor thesis philosophy economics students submitted received bachelor thesis summer semester 2012 scholarship semester abroad granted bavarian university centre chin				

5.6.1.1 Discussion on Vector Representation Models

A very large feature size was required for TF and TFIDF vector representations. In effect, this significantly increases the memory and time needed to train a prediction model, reducing its appeal to a practical application. On another observation, the sparsity of the feature vectors was above 0.99. This meant that most of the elements in the feature vectors were zero. On hindsight, sparse matrix compression formats could have been used to process the feature vectors and in turn reduce the computational requirements. All the models performed vector inference in similar timing. This meant that when it comes to online performance, all the models are suitable.

It was clear the results were equally poor across all approaches when D2V model was used. It appeared that they were performing at random capacity. This was further validated by comparing their performance with a dummy classifier that

simply classifies at random. Why did D2V, a state of the art model perform so badly in this problem? D2V is a sentence embedding model that considers word ordering during inference. Application content tends to be very short and when aggregated, the word order offers no information. Consequently, the vectors inferred from Applications and CV sections could end up very different in the vector space even though they shared similar context from a human's perspective.

The W2V vectors performed the best among all the vector representation models. It should be noted that the vectors were created by averaging the individual word vectors thus there were no word order information. Contrasting this to D2V, this showed that word order may not be as important in representing the CV sections and application forms, in fact could be detrimental. Another feature of W2V was its ability to hold semantic information of each word, which could explain its better performance over TF and TFIDF. The W2V model was thus selected to be the vector representation model of choice due to its smaller dimensions and much better performance over a wide range of approaches.

An example is shown in Table 5.4 to better illustrate the differing performance on the classification when using TFIDF, W2V and D2V. Given a sample graduate CV and its sections portrayed in Appendix B, the sections selected for the four questions using Inverse Answering by Similarity approach with TFIDF, W2V and D2V vector models are displayed. The table shows that with W2V, most of the answers make sense to humans. TFIDF did not do as well and D2V's answers are not comprehensible.

5.6.2 Comparison of approaches

Table 5.5 showed that without optimization, the Inverse Answering by Similarity approach scored an F1 score of 0.83 when trained on the Applications represented by W2V vectors of 100 dimensions. The Inverse Answering by Topic Modelling approach scored an F1 score of 0.85 based on almost 500,000 unique words in the corpus vocab. Table 5.6 shows a Stuart-Maxwell Test on the 2 sets of results and concluded that the difference in the performance here was statistically not significant, thus it is inconclusive that Topic Modelling perform better than Similarity

Table 5.4: An illustrated example of the vector model's effect on the SIM approach. Given a sample graduate CV and its numbered sections portrayed in Appendix B, the sections selected to answer the four questions using Inverse Answering by Similarity approach with TFIDF, W2V and D2V vector models are displayed. The table shows that with W2V, most of the answers make sense to humans. TFIDF did not do as well and D2V's answers are not comprehensible. Note that a human could easily relate sections 1 and 12 to answer 'Personal Details', section 2 to answer 'Skills', sections 3 to 6 to answer 'Education' and sections 7 to 11 to answer 'Work Experience'.

Question	TFIDF	W2V	D2V
Personal Details	1,5,6,12	1,12	
Education	3,4,7,8,9,10,11	3,4,6	1,4,6,7,8,9,10,11
Skills	2	2	2,3,5,12
Work Experience		5,7,8,9,10,11	

Table 5.5: F1 scores of the Inverse Answering by Similarity approach's prediction model on W2V, and Topic modelling based on TF with vocabulary size of approximately 500,000. The Inverse Answering by Topic Modelling approach appeared to perform slightly better than the Inverse Answering by Similarity approach. This was however deemed inconclusive after significance testing.

Approach	F1 score
SIM	0.83
TOPIC	0.85

approach.

For the Inverse Answering by Classification approach, Logistic Regression (LOG), Support Vector Machine (SVM), Multi-Layer Perceptron (MLP),Naive Bayes (NB) and Boosted Tree (XGB) depicted in Table 5.7 were experimented

Table 5.6: Was the Inverse Answering by Topic Modelling approach performing better than the Inverse Answering by Similarity approach by chance? Stuart Maxwell Test was performed here. A p-value of less than 0.05 rejects the null hypothesis that the results of the approaches are statistically not different. In this case, the p-value of 0.11 was much above 0.05 and thus it cannot reject the null hypothesis. Therefore, the better performance of one approach over the other was not conclusive

Approach	SIM	TOPIC
SIM		0.11
TOPIC	0.11	

Table 5.7: Supervised Learning Classifiers and their parameters used in the Inverse Answering by Classification approach.

Supervised Learning Classifier	Abbreviation	Remarks
Logistic Regression	LOG	One Vs Rest multiclass classification
Support Vector Machine	SVM	One Vs Rest multiclass classification with linear kernel
Multi Layer Perceptron	MLP	100 hidden layers with RELU activation function. Learning rate at 0.001.
Boosted Trees	XGB	XGBoost[41]
Nave Bayes	NB	Multinomial for TF and IDF, Guassian for W2V and D2V. Laplace smoothing

Table 5.8: F1 scores of Inverse Answering by Classification approach based on W2V vector models. LOG, SVM and MLP are the only classifiers that can perform at 0.7 and above, with LOG and SVM taking the lead. NB and XGB performed badly.

Approach	F1 score
LOG	0.768
SVM	0.766
MLP	0.697
NB	0.301
XGB	0.610

to assess their suitability from 3 perspectives, namely the ability to handle large dimensions, to perform prediction of CV Sections on a model trained with Application sections and lastly their suitability for practical applications.

Table 5.8 showed that without optimization, LOG, SVM and MLP classifiers performed with F1 scores of 0.70 and above when using W2V vector representations. Although the NB (Multinomial) prediction model performed with an F1 score of 0.7 using TF vector model as seen in Figure 5.6, it failed to perform with W2V vector model. The above observations were supported with the statistical tests in Table 5.9.

Next, each proposed approach's suitability in practical applications was analyzed. The similarity approach required the longest loading and prediction time

Table 5.9: P-values based on Stuart Maxwell Test or Wilcoxon's Sign Test if the former fails. A P-value of less than 0.05 rejects the null hypothesis that there is no significant difference between the results of the approaches. In this case, the pairwise performance of each classifier were statistically different.

Approach	LOG	SVM	MLP	NB	XGB
LOG					
SVM	5.919122e-03				
MLP	5.718074e-03	2.795150e-04			
NB	3.492750e-19	4.863374e-17	4.820434e-20		
XGB	2.773724e-02	3.952482e-04	1.305576e-03	2.771684e-17	

Table 5.10: The Inverse Answering by Similarity approach do not require any form of training prior to prediction but it took extended time to load the training set for cosine similarity computation at prediction later. The Inverse Answering by Topic Modelling and Inverse Answering by Classification approaches used CV Sections and Applications for training respectively but the loading time was much lower than the Inverse Answering by Similarity approach. In terms of prediction time, the Inverse Answering by Similarity approach incurred longer delays due to cosine similarity matrix computation

Approaches	Training	Loading time (sec)	Prediction time for 100 CV sections (sec)
SIMILARITY	Not required	54	5
TOPIC	CV Sections	5	<1
LOG	Application Forms	<1	<1
SVM	Application Forms	<1	<1
MLP	Application Forms	<1	<1

(based on 100 CV sections). The topic modelling approach required 5 seconds to load the model but this was acceptable considering that it held approximately 500,000 unique words in its model. All the Inverse Answering by Classification approach's classifiers performed the loading and prediction in less than a second each.

5.6.2.1 Discussion on comparisons of proposed approaches

The Inverse Answering by Similarity approach was the only approach that performed consistently better than other approaches across all vector representation models, except for D2V vector representation. The results show that the Inverse Answering by Similarity approach was on par with the Inverse Answering by Topic Modelling approach and perform better than the Inverse Answering by Classifica-

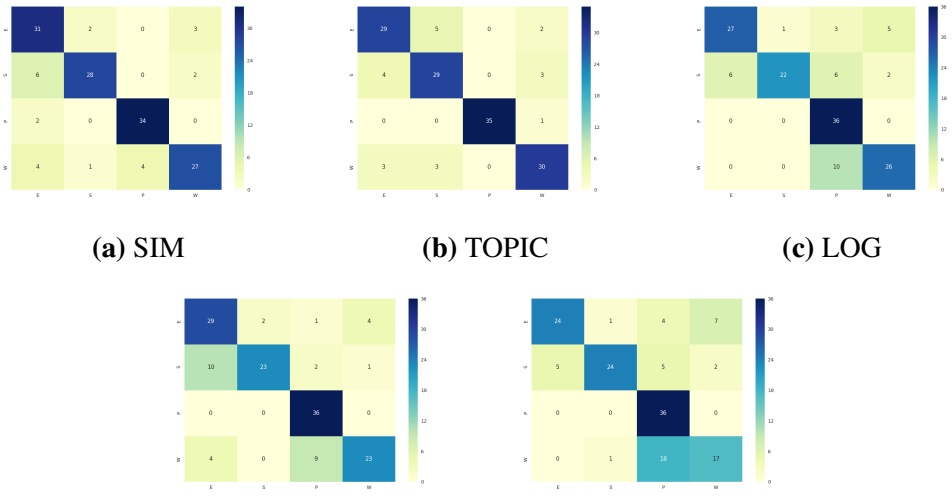
tion approach when using W2V. However, Table 5.10 showed that it was necessary to load the vectors of the training set during runtime so that a similarity function could be computed during prediction of unseen CV sections. This initial loading increased the time needed to initialize and did not appeal to practical applications. Despite this, it was possible to optimize this part of the implementation by reducing the training size. Therefore, this approach was retained as one of the approaches of choice for its good performance.

The Inverse Answering by Topic Modelling approach was on par with the Inverse Answering by Similarity approach and out-performed the Inverse Answering by Classification approach. It do not require any Applications and only require the CV sections as the training set for its topic model and thus made it very appealing to practical applications. However, the success of this approach was highly dependent on how well the CV sections were split and they must fulfill the assumption of being able to be distributed into at least the four topics being predicted. For its performance and fulfilled assumption in this dataset, this approach was retained as an approach of choice.

The Inverse Answering by Classification approach was experimented on several classifiers. The NB prediction model only performed well on the TF and TFIDF vector models. The XGB classifier performed but was unable to hit the minimum F1 score of 0.7. These classifiers were thus dropped from the approaches of choice. LOG, SVM and MLP classifiers remains the top 3 classifiers for the different vector models although their performance on TF vector model is inconclusive. For their performance, the LOG, SVM and MLP classifiers remain on the approach of choice for the remaining experiments.

5.6.3 Classification Performance

The confusion matrices of the approaches of choice in Figure 5.7 were used to analyze the performance of each class, that is, if the prediction of a particular class was especially good or bad or if a particular class was always misinterpreted as another class. The actual labels are displayed as rows while the columns represent predicted values. The classes are in order of 'Education', 'Skills', 'Personal Details'



(a) SIM **(b)** TOPIC **(c)** LOG

(d) SVM **(e)** MLP

Figure 5.7: The rows of confusion matrices are the actual values while the columns are the predicted values. The classes are in order of 'Education', 'Skills', 'Personal Details' and 'Experience'. Darker cell indicates a higher number. 'Personal Details' class was best classified for all approaches. The similarity and topic modeling approaches are more consistent with the classification across classes. The Inverse Answering by Classification approach's classifiers had problems with 'Experience' class, frequently misclassifying it as 'Personal Details'.

Table 5.11: F1 score of each class for each approach of choice. The approaches performed best on the 'Personal Details' class while the classification of 'Work Experience' class was the least performing among the classes. Despite that, all the classification achieves at least an F1 score of 0.70.

	Education	Skills	Personal Details	Work Experience
SIM	0.734	0.754	0.857	0.718
TOPIC	0.806	0.795	0.986	0.833
LOG	0.783	0.746	0.791	0.754
SVM	0.734	0.754	0.857	0.718
MLP	0.738	0.774	0.727	0.548

and 'Experience'. A darker cell indicates a higher number. 'Personal Details' was classified correctly most often for all approaches. The similarity and topic modeling approaches were consistent with the classification across classes. The performance varied for the classifiers of the Inverse Answering by Classification approach but the F1 score of each classes' classification were still above 0.7 in all cases as shown in Table 5.11.

5.6.3.1 Discussion on classification performance

All the approaches were able to classify the 'Personal details' class to a very high precision and recall which contributed to a high F1 score as seen in Table 5.11. 'Personal Details' class tend to come with very specific word patterns such as Addresses, Emails, and Phone numbers. This made the feature very discernible and would explain the higher performance in the classification of this class.

The proposed approaches performed to a satisfactory level with the 'Education', 'Skills' and 'Work Experience' classes as well although different approaches exhibit slightly different strengths in classifying them. Analyzing Figure 5.7, the Inverse Answering by Similarity and Inverse Answering by Topic Modelling approaches were very similar in terms of their misclassification distribution. For example, both approaches were confident that for a 'Education' and 'Skills' class, it would not be 'Personal Details'. For the Inverse Answering by Classification approach, LOG and MLP performed very similar in terms of misclassification distribution. SVM seemed to have it own unique set of classification characteristics.

In conclusion, 'Personal Details' was confidently classified by all approaches. Although the performance varies, the F1 score of each class's classification was still above 0.7. The approaches of choice could be organized into three groups based on their similar classification characteristics, this information would be useful in improving classification performance as groups of approaches rather than individual approaches.

5.6.4 Robustness of Heuristic Evaluation

To assess Heuristic Evaluation, the accuracy metrics were observed from Crowdsourced Evaluation to compare it with the accuracy metrics observed from Heuristic Evaluation. The reader should be reminded at this point that the Crowdsourced Evaluation used gold labels created from labelers. Figure 5.8 clearly shows that while Heuristic Evaluation couldn't represent the accuracy of individual approaches, it seemed to show a consistent relative performance of approaches. To validate this observation, the accuracy scores of every prediction model were taken from both Heuristic and Crowdsourced Evaluation and computed for a Pearson Cor-

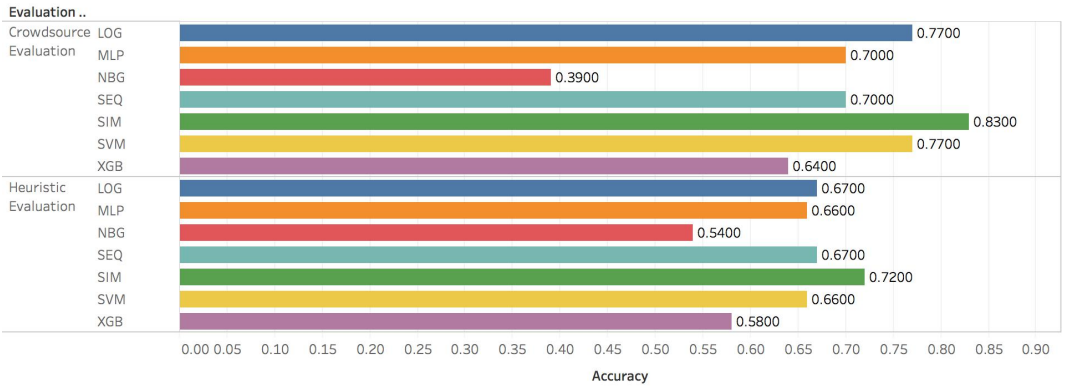


Figure 5.8: Heuristic Evaluation evaluate by seeking identical words between the CV Sections and their corresponding Applications, if these number of words exceeds a threshold, it will be a HIT, else a MISS. Thus, Heuristic Evaluation can only be evaluated via accuracy. The figure shows the accuracy results of both Heuristic and the gold standard Crowdsourced Evaluation. Heuristic Evaluation is not representative of the performance of each approach; however, a visual inspection shows they are representative of the relative performance of each approach.

relation index as follows.

$$P = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}))}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{5.7}$$

where n is the number of accuracy results from the different prediction models, x_i is the ith result from Heuristic Evaluation and y_i is the ith result from Crowdsourced Evaluation. The resulting Pearson Correlation index is 0.928. This implied a very strong positive correlation between the results of Heuristic and Crowdsourced Evaluation. This meant that Heuristic Evaluation is a strong representation of the relative performance of the different approaches. For example, if the Heuristic Evaluation showed that the Inverse Answering by Similarity approach is a better performer than the Inverse Answering by Classification , then it will likely be true for Crowdsourced Evaluation as well. The impact of this observation is that the Recruitment Agency do not need to perform expensive Crowdsourced Evaluation on large amount of data when they need to compare different prediction models.

5.6.4.1 Discussion of Heuristic Evaluation

The Heuristic Evaluation was susceptible to situations where information was found in CV but not found in the respective Applications. For example, an Application

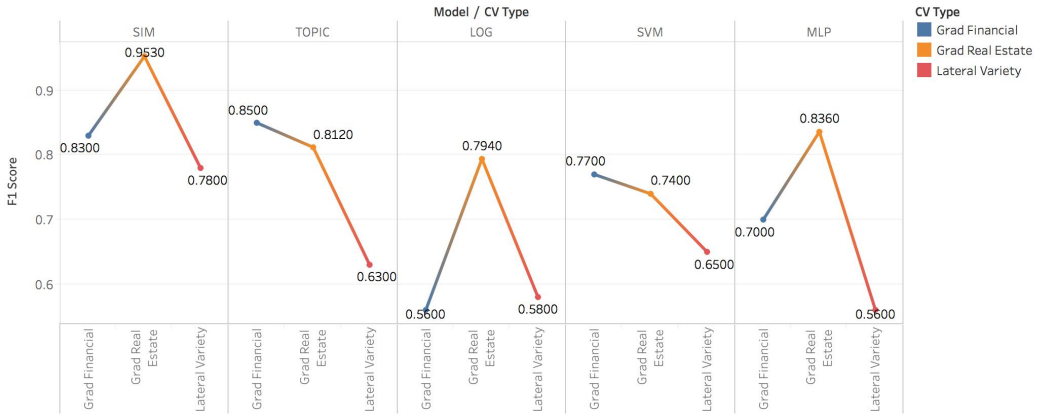


Figure 5.9: Comparison between results when tested with different domains and different variety of CVs. When tested in a different industry/domain (E.g. Finance to Real Estate), all approaches either perform better or have very little drop in performance. When tested on a wide variety of CVs, all but Inverse Answering by Similarity approach suffered a significant drop in performance.

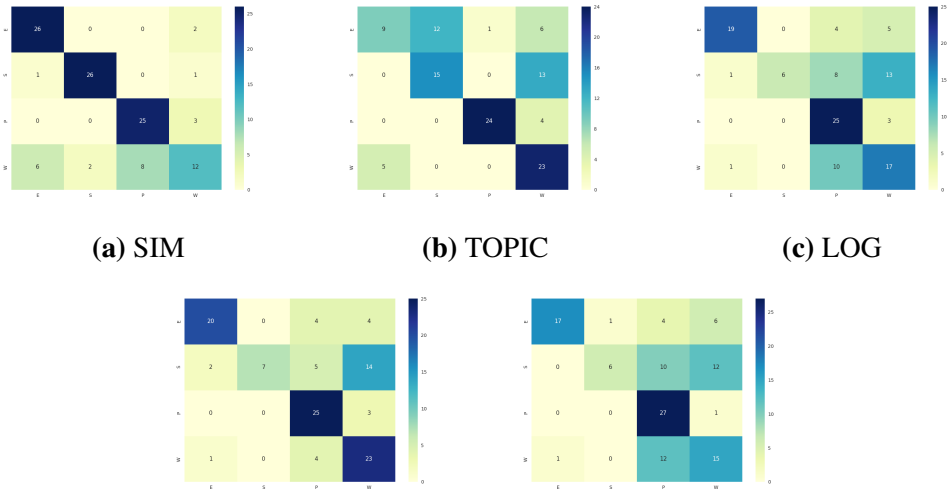
may not require the candidate to fill in 'skills', or the candidate left the field empty although required. This resulted in some MISSES when it should have been a HIT. This resulted in the generally lower accuracy scores for each prediction model.

5.6.5 Cross-domain effectiveness

The experiments so far focused on CVs from graduates applying for jobs in the financial industry. CVs from two very different domains were used to test the proposed approaches' cross domain effectiveness.

The first set consists of graduate CVs just as the initial set but this time they were taken from the real-estate industry. This set of CVs would test the performance for CVs from a different domain. The second set consists of lateral CVs, meaning the CVs came from a wide variety of candidates of different background and education levels. This set of CVs had a much higher variability in the content structure and test whether the CVs could perform in a much difficult scenario.

Figure 5.9 shows the change in performance of each approach of choice from Graduate Financial to Graduate Real Estate to Lateral Variety. When tested in different industry/domain, all approaches either performed better or had very little drop in performance. When tested with CVs from a wide variety of candidates, all but the Inverse Answering by Similarity approach suffered a significant drop in



(a) SIM **(b)** TOPIC **(c)** LOG

(d) SVM **(e)** MLP

Figure 5.10: Confusion matrices of the results when using a lateral variety of CVs. For similarity model, its slight performance drop was mainly due to the 'Work Experience' class. All approaches held well for the 'Personal Details' class but have varying results for the rest of the classes.

performance.

5.6.5.1 Discussion on cross-domain effectiveness

An analysis of the confusion matrices of each approach in Figure 5.10 showed that the performance on lateral variety CVs suffered mainly due to the higher misclassification rate in the 'Skills' and 'Work Experience' classes.

An observation of the lateral variety CVs was performed to understand the reason of the drop in performance, an example of such CV is presented in Appendix C. An observation was that the Lateral CVs are generally less organized and structured compared to the ones submitted by the graduates. This posed several problems. First, the CVs tend to be 'story based', as opposed to the structured chronological style provided by graduates. For example, there were cases where skills, study and work history are mentioned together in a single paragraph. The CV splitting algorithm was not robust against such CVs, as a result splitting larger sections of CVs that were more ambiguous. Even during the labeling process, the labeler noticed more difficulty labeling due to the ambiguity. With this observation, it was better understood why the approaches did not fare well for the classification of 'Ed-

Table 5.12: An illustrated example on how a less structured CV affected the results. Given a sample lateral CV and its numbered sections portrayed in Appendix C, the sections selected to answer the four questions using Inverse Answering by Similarity approach with TFIDF, W2V and D2V vector models are displayed. In this particular sample, it shows that the usually accurate 'Personal details' was predicted wrongly as 'Work Experience'. It should be noted that logically, section 1 would answer 'Personal Details', section 2 would answer 'Work Experience' and section 3 would answer 'Education'.

Question	TFIDF	W2V	D2V
Personal Details			
Education	3	3	2,3
Skills			
Work Experience	1,2	1,2	1

ucation', 'Skills' and 'Work Experience' for lateral CVs. The reader may refer to Appendices B and C for a random sample of the graduate and lateral CVs. Table 5.12 illustrates the weakness of the proposed approaches on lateral (less structured) CVs.

Conclusively, the effectiveness of the approaches held across domains but is very sensitive to the way the CVs were styled and eventually broken down.

5.6.6 Optimization

5.6.6.1 Effect of data -preprocessing

The results discussed previously already converted all content to lower case and included removal of stop-words and punctuation in a bid to reduce vocabulary size. Contiguous alphanumeric characters adjacent to non alphabets and numbers were also separated into individual words. For example, mymail@mail.com was splitted into 'mymail', 'mail' and 'com'. A couple more data preprocessing that was later performed were data flattening and lemmatization. For flattening, strings identified as numbers, emails, months, and years were flattened to be reflected as XXXX, WCNEMAIL, WCNMONTH, WCNYEAR respectively. The length of the numbers was retained as their length could mean different things. For example, XXXXXX could be a postal code while (XX)-XXXXXXX could be a phone number. By retaining the length, a more discriminative feature was produced.

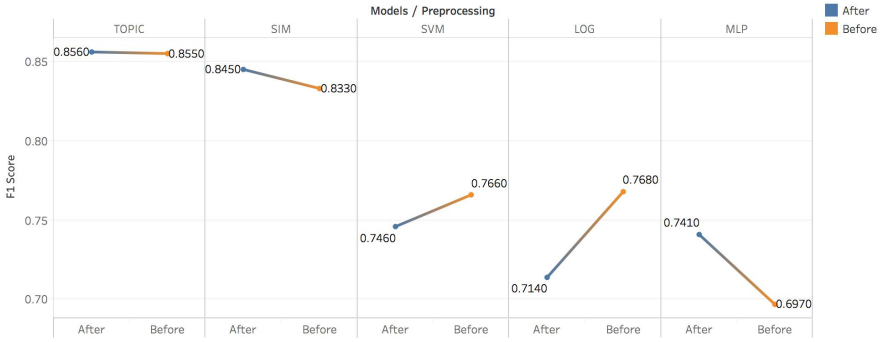


Figure 5.11: F1 scores of each approach before and after the data flattening and lemmatization. There were no strong impact on TOPIC and SIM models but improve SVM and LOG slightly. MLP suffered a slight dip in performance.

Instead of stemming, lemmatization was performed on the data before being fed to train the vector representation and approaches. Lemmatization group together inflected forms of a word, identified by the word's dictionary form. Compared to stemming, lemmatization is considered contextually more accurate.

On the Inverse Answering by Topic Modelling approach, the additional data flattening and lemmatization resulted in a 4-topic modelling's topic-word distribution that could not be confidently identified by humans. However, the 5-topic modelling produced a topic-word distribution that not only presents the four main classes, but also grouped school extra-curricular activities as the fifth topic. Table 5.13 shows the word distribution of the 5 topics. Based on this observation, CV sections that had highest probability for topic 1 were classified as 'Education' class as well.

Figure 5.11 shows the F1 scores of each approach before and after the data flattening and lemmatization. The result shows that there is no strong impact on TOPIC and SIM models but improve SVM and LOG slightly. MLP suffered a slight dip in performance.

Table 5.13: The top 20 most relevant words for each topic generated by LDA. By observation, topics 1,2,3,4,5 are identified as 'Work Experience, School Extra Curricular Activities (ECA)', 'Education', 'Personal Details' and 'Skills' respectively. Since ECA is considered schooling, CV sections that had highest probability for topic 2 were classified into 'Education' class.

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
wcnmonth	xxxx	xxxx	xx	skills
XXXX	wcnmonth	x	xxx	english
XX	student	wcnmonth	xxxx	interests
financial	team	xx	x	excel
company	xx	university	wcnemail	microsoft
X	member	school	xxxxx	fluent
research	present	finance	e	language
investment	school	business	wcnmonth	office
management	work	economics	email	c
analysis	university	education	address	computer
client	club	science	street	basic
intern	experience	management	mail	french
project	event	gpa	mobile	word
business	leadership	mathematics	road	native
experience	skill	financial	street	powerpoint
new	society	bachelor	xxxxxx	spanish
data	x	xxx	xxxxxxxxxx	proficient
team	xxx	international	ny	programming
analyst	year	college	phone	bloomberg
summer	wa	accounting	new	x

Chapter 6

Conclusion

Three approaches to answering four broad questions of applications forms via CVs have been implemented. The Inverse Answering by Topic Modelling approach is the most effective approach. Its independence from application forms and efficient running times makes it the most appealing approach to use in practical scenarios. The Inverse Answering by Similarity approach, when used with a W2V vector representation model performed consistently across different domains. It is on par with the Inverse Answering by Topic Modelling approach but its current implementation is not efficient enough for practical applications. Logistic Regression remains the better performing of the algorithms experimented in the Inverse Answering by Classification approach. Although its performance is less than that of the Inverse Answering by Similarity approach, its efficiency in the execution of the whole pipeline makes it a more practical approach to use. All the approaches perform best when answering 'Personal Details', but tend to misclassify 'Work Experience'. All the proposed approaches performed well on CVs from a different industry but did not fare well when tested on candidate CVs from a wide variety of background and education levels. Data preprocessing such as data-flattening and lemmatization did not offer much improvements.

A Heuristic Evaluation was proposed with the intent to perform unlabeled evaluation. The approach was not suitable for evaluating individual algorithms but was found to be very reliable in assessing the relative performance of different algorithms.

6.1 Future work

To significantly increase the robustness of the approaches, the splitting of CVs into sections could be improved. This was not within the scope of this thesis but should be looked into for future related work. This work has successfully answered the broad questions of recruitment Applications from CVs. This made it possible to refine the scope to answer specific questions such as age and gender. For example, having answered the broad question on personal details, it would be possible to extract age and gender information with less discrepancies than when its done on entire CVs.

Chapter 7

Appendices

Appendix A

Performance Metrics

A.1 Accuracy

Accuracy is defined by the number of correct predictions divided by the total number test examples, which makes it an easy metric to compute and understand. With a balanced test set, accuracy can be used effectively as a course gauge to the overall classification performance of each approach and their vector representation model. With reference to a binary confusion matrix in Table A.1, the formula is represented as

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \tag{A.1}$$

A.2 Precision, Recall and F1

Intuitively, in a binary classification, precision is defined as the ratio of correct positive predictions to the total number of positive predictions made. Recall is defined as the ratio of correct predictions to the all the data in that class. With

Table A.1: Binary class confusion matrix. The rows represent the gold labels while the columns represent the predicted classes

	A	B
A	True Positive (TP)	False Negative (FN)
B	False Positive (FP)	True Negative (TN)

Table A.2: Multi-class Confusion Matrix. The rows represent the gold labels while the columns represent the predicted classes

	Education	Skills	Personal Details	Work Experience
Education (E)	TP_E	E_{ES}	E_{EP}	E_{EW}
Skills (S)	E_{SE}	TP_S	E_{SP}	E_{SW}
Personal Details (P)	E_{PE}	E_{PS}	TP_P	E_{PW}
Work Experience (W)	E_{WE}	E_{WS}	E_{WS}	TP_W

reference to a binary confusion matrix, they are generally represented as

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{A.2}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{A.3}$$

To represent precision and recall as a single measure, the F1 score that represents the harmonic mean of the two metrics was used as shown in Equation A.4

$$\text{F1} = \frac{2 * (\text{Recall} * \text{Precision})}{\text{Recall} + \text{Precision}} \tag{A.4}$$

To present the F1 score for the overall classification instead of a single class, the F1 score for each class is averaged across.

$$\text{F1}_{avg} = \frac{\text{F1}_E + \text{F1}_S + \text{F1}_P + \text{F1}_W}{4} \tag{A.5}$$

where F1_E, F1_S, F1_P and F1_W represent the F1 scores for the classes 'Education','Skills','Personal Details' and 'Work Experience' respectively.

A.3 Multi-class Precision, Recall and F1

In a multi-class scenario, it is only required to compute the respective TN, TP, FN, FP values for each class and apply that to the formulation in Section 5.5.2. Table A.2 shows the multi-class confusion matrix of the problem here. The formulas to compute the TP_E,TN_E,FP_E,FN_E of the 'Education' class is shown from Equations A.6 to A.9.

Table A.3: 4x4 Agreement table for McNemar's test.A, B, C and D denoted Education, Skills, Personal Details and Work Experience respectively

	A	B	C	D	Row total
A	a	b	c	d	A.=a+b+c+d
B	e	f	g	h	B.=e+f+g+h
C	I	j	k	l	C.=i+j+k+l
D	m	n	o	p	D.=m+n+o+p
Col total	.A=a+e+i+m	.B=b+f+j+	.C=c+g+k+o	.D=d+h+l+p	

$$\mathrm{TP}_E = \mathrm{TP}_E \tag{A.6}$$

$$\mathrm{TN}_E = \mathrm{TP}_S + \mathrm{E}_{PS} + \mathrm{E}_{WS} + \mathrm{E}_{SP} + \mathrm{TP}_P + \mathrm{E}_{WS} + \mathrm{E}_{SW} + \mathrm{E}_{PW} + \mathrm{TP}_W \tag{A.7}$$

The above equation is intuitively the sum of all cells in the confusion matrix except those in the class's columns and rows.

$$\mathrm{FP}_E = \mathrm{E}_{SE} + \mathrm{E}_{PE} + \mathrm{E}_{WE} \tag{A.8}$$

The above equation is intuitively the sum of all cells in the confusion matrix within the class's columns, minus the TP of the class.

$$\mathrm{FN}_E = \mathrm{E}_{ES} + \mathrm{E}_{EP} + \mathrm{E}_{EW} \tag{A.9}$$

The above equation is intuitively the sum of all cells in the confusion matrix within the class's rows, minus the TP of the class.

A.4 Generalized McNemar's Test

McNemar's test work on pairwise agreement to determine if the performance difference is significant. This test can however only be performed on binary classifiers thus a generalized version of the McNemar's test is required. This generalized version is also known as the Stuart-Maxwell test. Given a pairwise approach, a 4x4 agreement table can be set up as shown in Table A.3. Out of N test samples, a represent the number of times when both Approach1 and Approach2 predicted class A, b represent the number of times when Approach1 predicted class B but Approach2

predicted class A and so on. Let column vector d contain any 4 of the values as depicted in Equations A.10 to A.14.

$$d = (dA, dB, dC, dD) \tag{A.10}$$

$$dA = A. - .A \tag{A.11}$$

$$dB = B. - .B \tag{A.12}$$

$$dC = C. - .C \tag{A.13}$$

$$dD = D. - .D \tag{A.14}$$

Let S denote the 4x4 matrix of the variances and co-variances of the elements of d. The elements of S are depicted in Equations A.15 to A.22

$$sAA = A. + .A - 2 * (a) \tag{A.15}$$

$$sBB = B. + .B - 2 * (f) \tag{A.16}$$

$$sCC = C. + .C - 2 * (k) \tag{A.17}$$

$$sDD = D. + .D - 2 * (p) \tag{A.18}$$

$$sAB = -(b + e) \tag{A.19}$$

$$sBC = -(g + j) \tag{A.20}$$

$$sCD = -(l + o) \tag{A.21}$$

$$sDA = -(m + d) \tag{A.22}$$

The Stuart-Maxwell statistic is calculated as:

$$X^2 = d' S^{-1} d \tag{A.23}$$

where d' is the transpose of d and matrix S^{-1} is the inverse of S. X^2 is interpreted as a chi-squared value with degree of freedom equal to 4 minus 1 equals 3. For each pair, we can either reject the null hypothesis at level α=0.05 and conclude that one approach is significantly better than the other, or accept the null hypothesis stating that the pair of approaches perform similarly on these data sets for level α=0.05.

Appendix B

Sample graduate CV

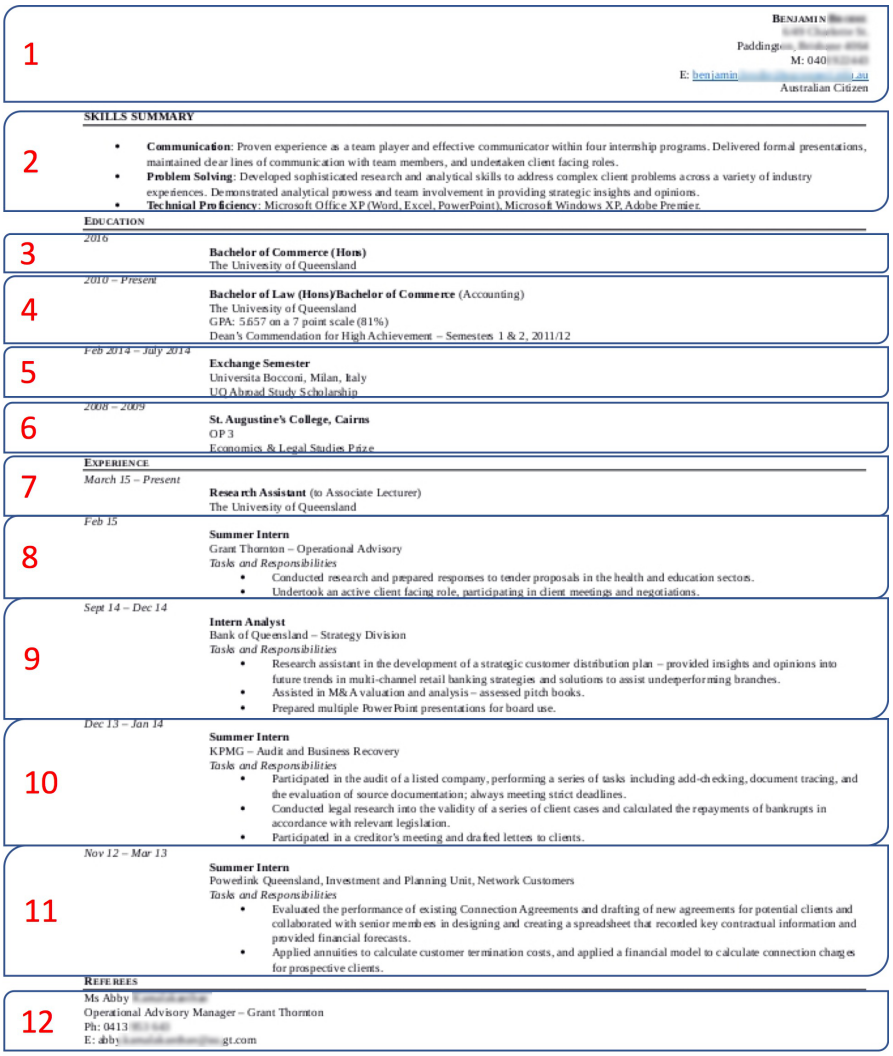


Figure B.1: Sample of graduate CV broken into sections by the splitting algorithm, numbered 1 to 12

Appendix C

Sample lateral CV

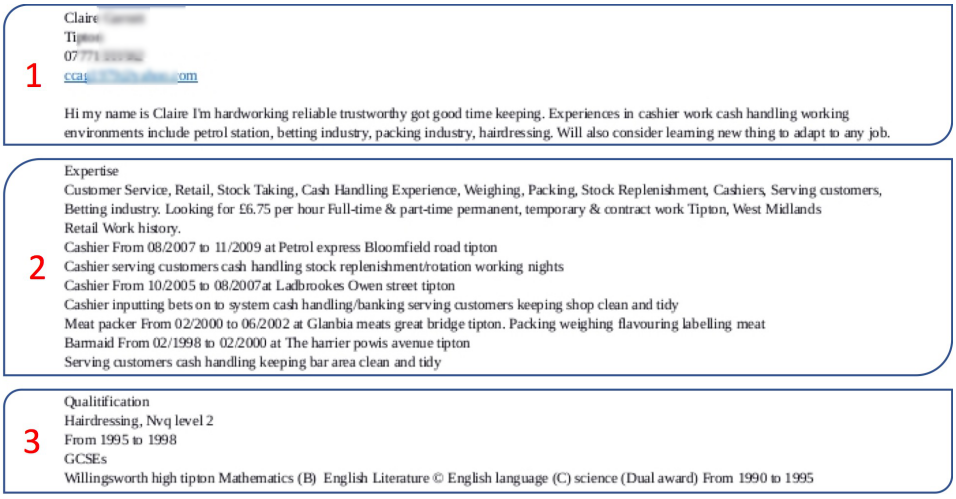


Figure C.1: Sample of lateral CV broken into sections by the splitting algorithm, numbered 1 to 3.

Appendix D

Source codes

The source codes can be viewed at GitHub.

`https://github.com/jax79sg/16073301_KahSiongTan_AnsAppForms`

The data used in this project, including the saved models are classified as confidential and will not be released into the public domain. Please contact the industrial partner for access to the data.

Bibliography

[1] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[2] François Chollet et al. Keras. `https://github.com/fchollet/keras`, 2015.

[3] Torsten Hothorn. Conditional inference procedures in a permutation test framework. `http://coin.r-forge.r-project.org/`. [Online; accessed 5-August-2017].

[4] Kah Siong Tan. Source codes to 'learning to answer recruitment application forms'. `https://github.com/jax79sg/16073301_KahSiongTan_AnsAppForms`, 2017. [Online; accessed Aug 30, 2017].

[5] Duygu elik. Towards a semantic-based information extraction system for matching resumes to job openings. *Turkish Journal of Electrical Engineering and Computer Sciences*, 24(1):141–159, January 2016.

[6] R. Mirizzi, T. D. Noia, E. D. Sciascio, and M. Trizio. A semantic web enabled system for r; composition and publication. In *2009 IEEE International Conference on Semantic Computing*, pages 583–588, Sept 2009.

[7] Dorin CARSTOIU Alexandra CERNIAN and Bogdan MARTIN. Semi-automatic tool for parsing cvs and identifying candidates abilities and com-

petencies. *2016 International Conference on Education, Management and Applied Social Science (EMASS 2016)*, 2016.

[8] Gunduka Rakesh Narsayya Momin Adnan Ayyas Prof. Khan Tabrez Mohd. Tahir Sayed Zainul Abideen Mohd Sadiq, Juneja Afzal Ayub. Intelligent hiring with resume parser and ranking using natural language processing and machine learning. In *International Journal of Innovative Research in Computer and Communication Engineering*, volume 4, April 2016.

[9] DERI Uldis Bojars. Resumerdf ontology specification. `http://rdfs.org/resume-rdf/`, 2013. [Online; accessed June 12, 2017].

[10] Melanie Tosik, Carsten Lygteskov Hansen, Gerard Goossen, and Mihai Rotaru. Word embeddings vs word types for sequence labeling: the curious case of cv parsing. In *VS@HLT-NAACL*, 2015.

[11] Chen Zhang, Hao Wang, and Yingcai Wu. Resumevis: A visual analytics system to discover semantic information in semi-structured resume data. *CoRR*, abs/1705.05206, 2017.

[12] Torsten Hothorn. Sovren, premier global provider of multi-lingual enterprise-grade resume and cv parsing and fourth-generation semantic searching and matching software. `https://www.sovren.com/`. [Online; accessed June 10, 2017].

[13] Textkernel,specialist in machine intelligence for matching supply and demand on the job market. `https://www.textkernel.com/company/about-textkernel/`. [Online; accessed June 10, 2017].

[14] Daxtra, intelligent, integrated, highly accurate, multilingual data capture, parsing, searching, matching and aggregation solutions. `http://www.daxtra.com/resume-database-software/resume-parsing-software/?gclid=Cj0KEQjwyZjKBRDu--WG9ayT_`

ZEBEiQApZBFuCjud69SqObxpoZWbncWF1jbN9WCzOwO6AuYlA_m9t4aAmXT8P8HAQ. [Online; accessed June 10, 2017].

[15] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *CoRR*, abs/1606.05250, 2016.

[16] Danqi Chen, Jason Bolton, and Christopher D. Manning. A thorough examination of the cnn/daily mail reading comprehension task. *CoRR*, abs/1606.02858, 2016.

[17] University of Illinois at Urbana-Champaign. Remedia story comprehension corpus. https://cogcomp.cs.illinois.edu/page/resource_view/11, 2013. [Online; accessed June 12, 2017].

[18] Hwee Tou Ng, Leong Hwee Teo, and Jennifer Lai Pheng Kwan. A machine learning approach to answering questions for reading comprehension tests. In *Proceedings of the 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora: Held in Conjunction with the 38th Annual Meeting of the Association for Computational Linguistics - Volume 13*, EMNLP '00, pages 124–132, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics.

[19] Yiming Cui, Zhipeng Chen, Si Wei, Shijin Wang, Ting Liu, and Guoping Hu. Attention-over-attention neural networks for reading comprehension. *CoRR*, abs/1607.04423, 2016.

[20] Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. *CoRR*, abs/1506.03340, 2015.

[21] Freebase. https://developers.google.com/freebase/. [Online; accessed June 14, 2017].

[22] Yahoo! answers. `https://uk.answers.yahoo.com/`. [Online; accessed June 14, 2017].

[23] Rana Malhas, Marwan Torki, Rahma Ali, Tamer Elsayed, and Evi Yulianti. Real, live, and concise: Answering open-domain questions with word embedding and summarization. In *TREC*, 2016.

[24] Guangyou Zhou, Tingting He, Jun Zhao, and Po Hu. Learning continuous word embedding with metadata for question retrieval in community question answering. In *ACL*, 2015.

[25] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[26] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[27] Bhuwan Dhingra, Hanxiao Liu, Ruslan Salakhutdinov, and William W. Cohen. A comparative study of word embeddings for reading comprehension. *CoRR*, abs/1703.00993, 2017.

[28] Jeff Mitchell and Mirella Lapata. Composition in distributional models of semantics. *Cognitive Science*, 34(8):1388–1429, 2010.

[29] Quoc V. Le and Tomas Mikolov. Distributed representations of sentences and documents. *CoRR*, abs/1405.4053, 2014.

[30] Hamid Palangi, Li Deng, Yelong Shen, Jianfeng Gao, Xiaodong He, Jianshu Chen, Xinying Song, and Rabab K. Ward. Deep sentence embedding using the long short term memory network:analysis and application to information retrieval. *CoRR*, abs/1502.06922, 2015.

[31] John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. Towards universal paraphrastic sentence embeddings. *CoRR*, abs/1511.08198, 2015.

[32] Portable document format (pdf) to text converter (version 3.00). `https://linux.die.net/man/1/pdftotext`. [Online; accessed June 14, 2017].

[33] A pure python-based utility to extract text and images from docx files. `https://pypi.python.org/pypi/docx2txt`. [Online; accessed June 14, 2017].

[34] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[35] Carson Sievert and Kenneth E. Shirley. "ldavis: A method for visualizing and interpreting topics". *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*, pages 63–70, June 2014.

[36] Landis JR and Koch GG. The measurement of observer agreement for categorical data. *Biometrics*, pages 159–74, March 1977.

[37] Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, Jun 1947.

[38] Alan Stuart. A test for homogeneity of the marginal distributions in a two-way classification. *Biometrika*, 42(3/4):412–416, 1955.

[39] A. E. MAXWELL. Comparing the classification of subjects by two independent judges. *The British Journal of Psychiatry*, 116(535):651–655, 1970.

[40] Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945.

[41] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *Ann. Statist.*, 29(5):1189–1232, 10 2001.