Thèse n° 8998

Detecting Latent Training Needs Using Large Datasets

Présentée le 23 juin 2022

Faculté informatique et communications
Laboratoire d'ergonomie éducative
Programme doctoral en informatique et communications

pour l'obtention du grade de Docteur ès Sciences

par

Ramtin YAZDANIAN

Acceptée sur proposition du jury

Dr D. Gillet, président du jury
Prof. P. Dillenbourg, Prof. R. West, directeurs de thèse
Prof. R. Lalive, rapporteur
Prof. N. Aljohani, rapporteur
Prof. K. Aberer, rapporteur

École
polytechnique
fédérale
de Lausanne

2022

Who dares to strike today, when having the security of
a permanent job is itself becoming a privilege?
— Slavoj Žižek

To Zahra, Mom, and Dad

Acknowledgements

This thesis is the result of a long and at times arduous journey, and it would not have been possible without the help and support of my supervisors, my family, and my friends.

There are many people to thank, but the first and foremost is Pierre. Working with you and learning from you has been an incredibly enlightening experience, and I have learned many valuable lessons from you that will guide me throughout the rest of my career. Thank you for your support and patience. I would also like to sincerely thank Bob for essentially introducing me to data science. The opportunity to work with you has been a great pleasure. The second half of this thesis was greatly improved through the invaluable input of Richard, whom I would like to thank deeply for his support throughout the last one and a half years of my PhD. Although we only worked together for a rather short period of time, the resulting impact on my thesis was immense, for which I am eternally grateful.

I would like to express my gratitude to Florence, thanks to whom none of us ever have to worry about administrative matters. I would like to thank all of my friends, especially Teresa, Kevin, and Joseph, for making early mornings (by which I mean 10 AM) a lot more fun than they would have been otherwise. I would additionally like to thank Louis, Sina, Utku, Jauwairia, Hala, Barbara, Aditi, Jennifer, Thibault, Arzu, Sven, and all other past and present CHILI members whom I have had the pleasure of knowing.

Last but absolutely not least, I have to thank the three people to whom I have dedicated this dissertation: my beautiful and loving wife Zahra, my mom, and my dad. Mom, dad, without your love and unwavering support, I would not have been here in the first place. Zahra, you are the most incredible person I have ever had the fortune of knowing, and I would not have been able to finish this dissertation without you. Thank you for being there for me.

Lausanne, February 9, 2022 R. Y.

Abstract

In today's world, there is no shortage of disruptors acting on various professional domains. The Fourth Industrial Revolution, with its AI-driven and automation-focused technologies, has fundamentally changed many domains – particularly the Information and Communication Technologies (ICT) domain – and continues to do so. The COVID-19 pandemic and its ensuing lockdowns have resulted in dramatic change in the workplace. These disruptors are fast-acting and rapidly change the skills landscape of many professional domains, introducing new skills and rendering old ones obsolete. In such a situation, it is imperative for educational institutions to keep their curricula updated in order to (re)train the workforce with the right set of skills for the new economy. However, most of the methodologies commonly used for analyzing training needs and effecting curricular change were developed during a time when the pace of change was slower, and thus have trouble quickly and effectively responding to today's rapid changes. As a result, the development of new approaches for the analysis of training needs is of utmost importance.

The present dissertation is the result of an effort to develop such new approaches, using different types of data and approaching the problem from different perspectives. These approaches use big data methods on data sources that are large, pre-existing, and continually updated in order to identify the skill needs of the present and/or predict the skill needs of the future for various professions. Focusing on the ICT domain but also making forays into vocational domains, the present work examines the usefulness and feasibility of methodologies such as the estimation of topical difficulty, the prediction of the appearance of new skills, and the prediction of emerging skills (less popular skills expected to experience a surge in hiring demand) for the identification of training needs. The findings of this dissertation are twofold. Firstly, they demonstrate the pivotal role that decentralized educational platforms such as Stack Overflow (an online Q&A platform for ICT) and Udemy (a MOOC platform where anyone can create a course) play in lifelong learning, essentially making the case for a policy of sponsorship and promotion of such platforms in other professional domains. Secondly and more importantly, the results showcase the feasibility of predicting the emerging skills of the near future. The successful emerging skills methodology can therefore be used to provide valuable insights to training providers at a moment's notice, giving them time to prepare for the skills landscape of the future in advance. Thus, AI can be used to help solve the challenges it creates for educational institutions and the workforce. Future work, both in the form of policymaking and in the form of research, can help capitalize and expand on the results of this dissertation, furthering the cause of rapid curricular change.

Keywords: training needs analysis, big data, lifelong learning, machine learning, skill needs, emerging skills, vocational education and training, MOOCs, Stack Overflow, job ads

Résumé

Dans le monde d'aujourd'hui, nombreux sont les élements qui perturbent le monde professionnel. La quatrième révolution industrielle, avec ses technologies axées sur l'IA et l'automatisation, a fondamentalement changé (et continue de changer) de nombreux domaines - en particulier celui des technologies de l'information et de la communication (TIC). La pandémie de COVID-19 et les confinements qui en découlent ont entraîné des changements spectaculaires du lieu de travail. Ces perturbateurs agissent rapidement et changent rapidement la perspective des compétences de nombreux domaines professionnels, introduisant de nouvelles compétences et rendant les anciennes obsolètes. Dans une telle situation, il est impératif que les établissements d'enseignement maintiennent leurs programmes à jour afin de (re)former la main-d'œuvre avec le bon ensemble de compétences pour la nouvelle économie. Cependant, la plupart des méthodologies couramment utilisées pour analyser les besoins de formation et effectuer des changements curriculaires ont été développées à une époque où le rythme du changement était plus lent, et ont donc du mal à répondre rapidement et efficacement aux changements rapides d'aujourd'hui. En conséquence, le développement de nouvelles approches pour l'analyse des besoins de formation est de la plus haute importance. La présente thèse est le résultat d'un effort visant à développer de telles nouvelles approches, en utilisant différents types de données et en abordant le problème sous différents angles. Ces approches utilisent des méthodes de big data sur des sources de données volumineuses, préexistantes et continuellement mises à jour afin d'identifier les besoins de compétences du présent et/ou de prédire les besoins de compétences de demain pour différents métiers. Se concentrant sur le domaine des TIC mais faisant également des incursions dans la formation professionnelle, le présent travail examine l'utilité et la faisabilité de méthodologies telles que l'estimation de la difficulté d'un contenu textuel, la prédiction de l'apparition de nouvelles compétences et la prédiction de compétences émergentes (compétences moins populaires, mais qui auront une forte augmentation de la demande) pour l'identification des besoins de formation. Les conclusions de cette thèse sont doubles. Premièrement, cette thèse démontre le rôle central que les plates-formes éducatives décentralisées telles que Stack Overflow (une plate-forme de questions/réponses en ligne pour les TIC) et Udemy (une plate-forme MOOC où n'importe qui peut créer un cours) jouent dans l'apprentissage tout au long de la vie, plaidant essentiellement pour une politique de financement et de promotion de telles plateformes dans d'autres domaines professionnels. Deuxièmement et surtout, les résultats montrent qu'il est possible de prédire les compétences émergentes dans un avenir proche. La méthodologie des compétences émergentes peut donc être utilisée pour fournir des informations précieuses aux prestataires de formation à tout moment, leur donnant le temps de se préparer pour les compétences de l'avenir. Ainsi, l'IA peut être utilisée pour aider à résoudre les défis qu'elle crée pour les établissements d'enseignement et pour l'emploi. Les travaux futurs, à la fois sous la forme d'élaboration de politiques et sous forme de recherche, peuvent aider à capitaliser et à développer les résultats de cette thèse, favorisant ainsi le processus d'évolution rapide des programmes d'études.

Contents

Acknowledgements **i**

Abstract (English/Français) **iii**

1 Introduction **1**

1.1 Motivation and context 1

1.2 Objectives and overview of contributions 2

1.3 Thesis roadmap 5

2 Research Overview and Related Work **7**

2.1 Review of related work 7

2.1.1 Training Needs Analysis: The What and the Why 7

2.1.2 The methods of training needs analysis 9

2.1.3 Big data, skill needs, and lifelong learning 12

2.2 Research overview 15

2.2.1 Study 1: Topical Difficulty on Stack Overflow 18

2.2.2 Study 2: Dynamics of Online Hiring and Learning Platforms 18

2.2.3 Study 3: Emerging Skills in ICT 19

2.2.4 Study 4: Emerging Skills in VET 19

2.2.5 Other studies 19

3 Study 1: Topical Difficulties on Stack Overflow **21**

3.1 Introduction 21

3.2 Related work 23

3.2.1 Topical difficulty 23

3.2.2 Online Q&A: Stack Overflow 24

3.3 Methodology 25

3.3.1 Topical difficulty methodology 26

3.3.2 Developer survey design 29

3.3.3 Research questions 32

3.4 Results 32

3.4.1 Topics and question response times . . . 32
3.4.2 Developer survey results . . . 35
3.5 Discussion . . . 38
3.5.1 Implications of results . . . 38
3.5.2 Refinements and future work . . . 39
3.6 Conclusions . . . 40

4 Study 2: Dynamics of Online Hiring and Learning Platforms 41
4.1 Introduction . . . 41
4.2 Related Work . . . 44
4.2.1 The role of Big Data in the new economy . . . 44
4.2.2 Big Data sources: online hiring and learning platforms . . . 45
4.2.3 Positioning the study . . . 47
4.3 Research methodology . . . 47
4.3.1 Platforms and datasets . . . 47
4.3.2 Aims and Methods . . . 51
4.3.3 Research questions . . . 58
4.4 Results . . . 59
4.4.1 Appearance Ordering . . . 59
4.4.2 Delays and Prediction Windows . . . 63
4.4.3 Tag themes and types . . . 69
4.5 Discussion . . . 71
4.5.1 Implications of results . . . 71
4.5.2 Generalizability . . . 73
4.5.3 Other MOOC platforms . . . 74
4.5.4 Threats to validity . . . 75
4.5.5 Future work . . . 76
4.6 Conclusions . . . 77

5 Study 3: Emerging Skills 79
5.1 Introduction . . . 79
5.2 Related work . . . 80
5.2.1 The necessity of curricular change in ICT . . . 80
5.2.2 Big data and skill landscapes . . . 81
5.3 Objectives and methodology . . . 82
5.3.1 Data and definitions . . . 82
5.3.2 Classification pipeline . . . 86
5.3.3 Research questions . . . 88
5.4 Results . . . 89
5.4.1 Predictive performance . . . 89

5.4.2 The limits of predictive performance 93
5.4.3 Important features 94
5.5 Discussion 95
5.5.1 Implications for educational institutions 95
5.5.2 Limitations 95
5.5.3 Future directions 96
5.6 Conclusion 98
5.7 Acknowledgements 98

6 Study 4: Emerging Skills in VET 99

6.1 Introduction 99
6.2 Related work 99
6.2.1 VET and Switzerland 99
6.3 Objectives and Methodology 101
6.3.1 Study Objectives 101
6.3.2 Data 101
6.3.3 Methodology 103
6.4 Results 104
6.4.1 Emerging skill classification 104
6.4.2 Investigating the reduced performance 105
6.4.3 Improving classifier performance 108
6.5 Discussion 109
6.5.1 Why the classifier failed 109
6.5.2 Emerging skills still matter 110
6.5.3 Future work 110
6.6 Conclusion 111

7 Discussion 113

7.1 Summary of results 113
7.2 Contributions 114
7.2.1 Emerging skills: the present and the future 114
7.2.2 Decentralized platforms for education 117
7.3 Future work 119
7.4 Conclusions 120

A Features used in the difficulty estimation classifier, Study 1 121

B Matching Stack Overflow tags to their occurrences in text, Study 2 123

C Signals of tag importance, Study 2 125

D Tag themes and types in Study 2 127

E Importance scores for emerging skill prediction features, Study 3 129

Bibliography 131

Curriculum Vitae 141

1 Introduction

1.1 Motivation and context

In today's world, innovative technologies are being developed at a rapid pace. The Fourth Industrial Revolution (Schwab, 2017) is in full swing, and many industries have been substantially disrupted – especially by skills that fall under the Information and Communication Technologies (ICT) umbrella (Michaels et al., 2014). Such innovative technologies, be they new products or new production processes (Gopalakrishnan and Damanpour, 1997), can (and often do) bring with them new skills that the workforce needs to learn (Kim, 2002). As a result, lifelong learning has emerged as a crucial element of the new economy in virtually every professional domain (Field, 2000), and the Swiss Vocational Education and Training (VET) system is no exception to this rule. Governments, educational institutions, and corporations all react to these changes to the skills landscapes (Prinsley and Baranyai, 2015; BGT, 2019; LinkedIn, 2019; Coursera, 2019). For educational institutions in particular, this translates into a need to speed up their "curricular change" processes, so that they can keep up with the pace of changes in each industry. In other words, they need to make sure that whether it's training or retraining, they are satisfying the present and future "training needs" of their students.

The phrase "training need" can be defined in several, broadly similar ways, but one particularly useful definition is that a training need is a discrepancy between the skills or knowledge that workers (in a particular organization or profession) need or the performance expected of them, and those that they currently have or exhibit (Gould et al., 2004). In other words, training needs are the skills and knowledge that the workers' training has not appropriately equipped them with. Training needs analysis (TNA) is the systematic collection and analysis of data leading to the detection of training needs, and is followed by the creation of a training program, and finally the evaluation of its efficacy, as part of a cyclic process. Broadly speaking, TNA can be conducted on one the four hierarchical levels of the profession, the organisation, the department, or the individual (Gould et al., 2004) (only the first two of which are relevant to our work), and is often done through questionnaires, surveys, interviews, and focus groups, although methodologies using economic census data and/or labor market data are also exist (ILO and OECD, 2018). Using these tools, information about current skills and needed skills is collected from workers, experts, or market trends, helping identify the gaps. In particular, in the Swiss VET system, the State Secretariat for Education, Research, and Innovation (SERI) and the Swiss Federal Institute for Vocational Education and Training (SFIVET) work closely with the cantons, professional organisations (representing Swiss businesses), and educational institutions in order to keep training curricula (called "ordonnances de formation professionnelle" in French) updated. For every profession, a committee is formed at the federal level to assess the need for curricular change at least once every five years, and professional associations can request changes to the curricula if they deem it necessary based on their industry knowledge (SKBF and CSRE, 2018). However, with the coincidence of an industrial revolution (Schwab, 2017) and a global pandemic (Bai et al., 2020), these methods of assessing the need for curricular change and the associated data collection and analysis methods may be too slow to keep up with the pace of change, especially for larger scopes involving entire professions and labor markets.

Big data can help speed up this process, acting as a transformative force in today's economy (Horton and Tambe, 2015; Einav and Levin, 2014): labor market intermediaries, such as online hiring platforms (i.e. job ad websites), Q&A forums, and online course platforms continuously generate large online datasets. These datasets provide views into the labour market with finer granularity, more rapidity, and potentially larger scopes than ever before, compared to administratively curated data. The latter allow for broad and multi-faceted analyses of employment patterns and large-scale changes in different sectors, but are hard and time-consuming to collect (and therefore infrequently collected) (Horton and Tambe, 2015). Many online learning platforms are particularly geared towards lifelong learning (Fischer, 2000; Latchem, 2016; Buhl and Andreasen, 2018), which is the type of learning required in this new, rapidly evolving economy. Job ad platforms such as LinkedIn, Q&A platforms such as the Stack Exchange family (especially Stack Overflow, which deals with software programming), and MOOC platforms such as Coursera and Udemy are all part of this emerging set of data sources (Horton and Tambe, 2015), and with the growing digitization of every industry, these platforms continue to rise in importance.

1.2 Objectives and overview of contributions

The present dissertation presents a series of approaches designed to use "big data", with **large**, **pre-existing**, and **continuously-updated** datasets to **identify present training needs or predict future training needs on the profession level**. The aim is to provide insights on the changing skills landscapes to educators and to help quicken the pace of curricular change.

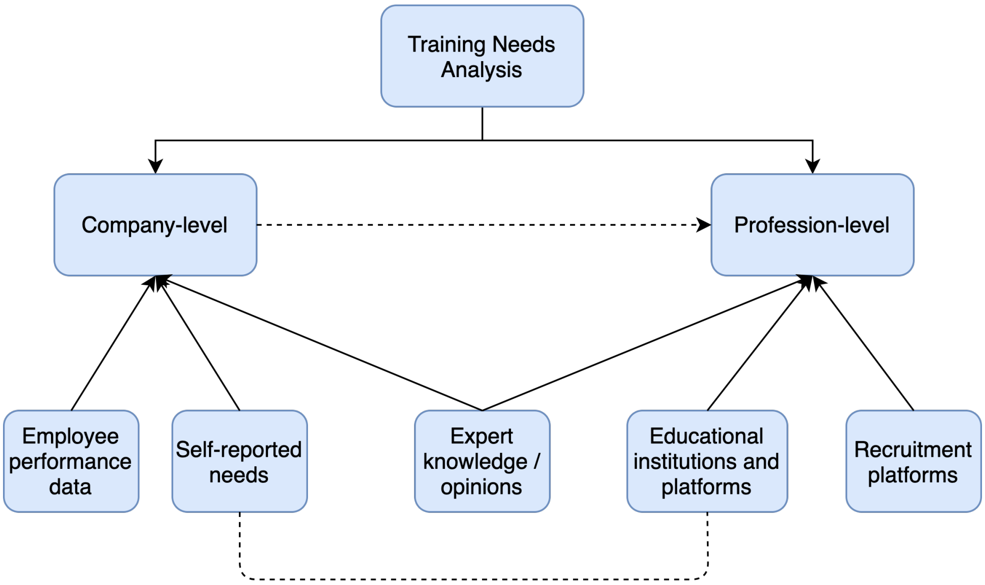


Figure 1.1 – The two levels of TNA relevant to the present work, along with the data sources that can be used for it.

The three attributes of the data sources used in the present work help alleviate the problems posed by traditional TNA approaches:

- **Large** datasets can provide much larger scopes and/or much finer granularities, therefore fitting our aim of conducting profession-level TNA.
- The dataset being **pre-existing** means that no time needs to be spent collecting the data in the first place, significantly quickening the TNA process.
- The **continuous updating** of the dataset (which would happen through the nature of the platform generating the data, rather than any action on the researcher's part) means that the analyses can be conducted at a moment's notice and will be up-to-date for the labour market at that point in time.

The top-level research question of this dissertation, which translates into several more specific questions in each research approach, is as follows: **Is an early identification of present training needs or an early prediction of future training needs possible, and if so, how early?**

Given the aim of speeding up curricular change, the two levels of TNA shown in Fig. 1.1 are of interest: company-level, and profession-level. The data sources of interest include both large-scale sources such as those from educational and hiring platforms or employee performance at companies, and survey-based data such as expert opinions and self-reported needs. These data sources enable either a direct approach to profession-level TNA, or an indirect approach based on aggregating company-level needs for several companies. Due to the near-prohibitive difficulty of acquiring internal data from multiple companies and aggregating their company-level training needs, the primary focus for most of this dissertation is on direct profession-level approaches, and there will only be a brief discussion of some company-level approaches that were tried. In any case, as the present work is based on digital data, the professional domains where the methodologies are developed would have to be heavily digitized. This is why most of the studies in the present work are conducted on the Information and Communication Technologies (ICT) domain using methodologies that are developed with both agility and generalizability in mind, while the final study deals directly with VET. The steady growth of digitization among most professional domains makes this a reasonable approach, as what is only available for ICT now may soon be available for many other domains (Schwab, 2017).

The research contributions of this dissertation are as follows:

1. The dynamics between online learning and hiring platforms are investigated and quantified in the software programming domain, showing that decentralized online learning platforms are quicker at manifesting new skills than job ads, and the more decentralized the platform, the quicker it is. It is also shown that predicting the **appearance** of **new topics** in job ads using signals from the learning platforms is impractical in the software programming domain, due to the fact that the delay in appearance is often no more than half a year, and the signals that appear in this period are not discriminative enough.
2. Finding the prediction of new topic appearance impractical, the idea of **emerging skills** is conceptualized as "skills with previously low hiring demand that have recently experienced a surge in hiring demand based on job ads". The hypothesis is that the past job ad trends of each skill could be used to predict its emergence or non-emergence in the future. Putting this hypothesis to the test in the ICT domain shows that such a prediction is possible. This is, therefore, a methodology that allows training providers to forecast the changes to the skills landscape in the near future. The fact that this approach only uses job ad data (and makes use of no auxiliary data sources) and only considers each skill in isolation (instead of considering the relationships between different skills) and still succeeds in predicting skill emergence means that there is considerable potential in expanded versions of this methodology.
3. Building on the success of the emerging skills approach in the ICT domain, the same approach - which is generalizable to any domain by virtue of its use of job ads as the sole data source - is applied to two VET domains, helping develop heuristics and feasibility checks that assess whether or not such a predictive approach would work in a particular domain.

In addition to demonstrating the agility of decentralized educational platforms in a rapidly evolving skills landscape and showing their immense potential for research on skill needs, the results of the present dissertation offer a cohesive two-step methodology for tackling the problem of skill need identification in any digitized professional domain. In the first step, the dynamics between educational and hiring platforms in the domain (and the internal dynamics of each) are analyzed in order to find out the speed at which the domain's skills landscape evolves, and the agility with which educational institutions respond to this evolution. This also results in a general understanding of the time scales involved in the domain, particularly the amount of time required for the creation of training content for that domain. In the second step, past job ad data (potentially combined with other, auxiliary data sources) are used to predict the emerging skills of the near future. These emerging skills are those which may have not been on educators' radars before (due to their lower previous popularity), but which *should be*, as they are expected to experience a quick rise in popularity. If the prediction task is successful within the confines of the time scales established in the first step (e.g. if it is possible to accurately predict 3 months in advance, *and* courses can be created in less than 3 months), then its results can inform educators of the skills that are soon to rise in importance, while allowing enough time for them to react to this information by creating training material for those skills.

The contributions listed above do not cover all the work done within this dissertation. Several less successful approaches were tried, which, although not yielding useful methodologies per se, ruled out some avenues of research and guided the research path towards more promising ones. An overview of all of these research approaches and the path taken through them will be given in the next chapter.

1.3 Thesis roadmap

The present dissertation is structured as follows:

- **Chapter 2** begins with a discussion of the previous literature that is relevant to this dissertation as a whole, and involves a deeper dive into the various data sources that have been used or can be used for training needs analysis. Then, the research path will be discussed and the different approaches that were taken in order to tackle training needs using Big Data will be explored. This discussion will illuminate the successes and failures of each approach, and how they informed the design of the next.
- **Chapter 3** lays out the details of Study 1, which centers around estimating the relative "difficulty" of software programming topics using question response times on Stack Overflow, which is a massively popular Q&A website in the software programming domain. This chapter deals with the design of the topical difficulty estimation pipeline,

the design of the survey given to software engineering practitioners in order to verify the topical difficulty estimates, and the results of this study. At the end of this chapter, the potential fixes that were envisioned to improve this study will be discussed, and the reasons for ultimately moving away from this approach will be laid out.

- **Chapter 4** deals with Study 2, which analyzes the dynamics of the appearance of new topics on online hiring and educational platforms in the software programming domain. This study analyzes Stack Overflow, Udemy (a MOOC platform), Stack Overflow Jobs (a hiring platform), and Google Trends (which shows Google search trends). The goals of the study, which involve both descriptive (i.e. understanding the dynamics of the first appearances of topics) and predictive aims (predicting the appearance of a topic on one platform using data from another), are discussed, and its results regarding both aims are presented. Then, the generalizability of this approach will be discussed, and the ways in which it informed the design of the next study are explained.

- **Chapter 5** deals with Study 3, which presents a classification methodology for predicting "emerging skills", which were defined in the previous section. The chapter will begin by discussing how emerging skills were defined based on the takeaways from the previous chapters, and then the design and results of the predictive pipeline are laid out. The performance of the predictive models is then analyzed in detail, and multiple avenues for the improvement of the methodology are discussed.

- **Chapter 6** deals with Study 4, which is the generalization of the emerging skills methodology from Study 3 to two VET domains: logistics, and healthcare. The central question in this chapter is whether the emerging skills methodology generalizes to these VET domains, and how the dynamics of the skills landscapes of these domains compare to that of the ICT domain, which is the main focus of the present dissertation. In addition, the results of this chapter are used to develop heuristics for checking the feasibility of emerging skills prediction in domains that are slower to evolve compared to ICT.

- **Chapter 7** is the final chapter, where all the findings of the present work are summarized and a synthesis of all the contributions is presented. The implications of the results for the original research question and motivations are discussed, showing the ways in which future work could build upon the work in this dissertation.

2 Research Overview and Related Work

The present dissertation aims to answer the following top-level research question: **Is an early identification of current profession-level training needs or an early prediction of future training needs possible, and if so, how early?** This question puts the present work at the intersection of multiple academic domains. Training needs analysis (TNA) is often categorized under Human Resources (HR) Management or Education, while analyses of big data for similar purposes often fall under the umbrella of Labor Economics. As a result, previous work relevant to the present dissertation is very diverse. In addition, a significant portion of the previous work done on the subject is by corporations and governmental organizations rather than academics, as we will soon see. In order to present these works in an organized fashion, I will present them from two perspectives: whether their focus is company-level or profession-level, and what data sources they focus on. As mentioned before, these data sources are not necessarily “big data”, but a thorough understanding of how all kinds of data are used for training needs analysis is pivotal to this dissertation. In this chapter, I will provide an overview of these previous works, followed by an overview of our own research path and how each approach succeeded or failed and how it informed the design of the next approach. After this overview, I will take a deeper dive into the specific data sources that my approaches have used. Some of the details on the literature and data sources specific to each study are reserved for their respective chapters.

2.1 Review of related work

2.1.1 Training Needs Analysis: The What and the Why

Training needs analysis, which is alternatively known as skill needs analysis or identification, deals with identifying the gaps between the skills and knowledge that workers have, and the skills and knowledge that they need for the satisfactory performance of their job functions

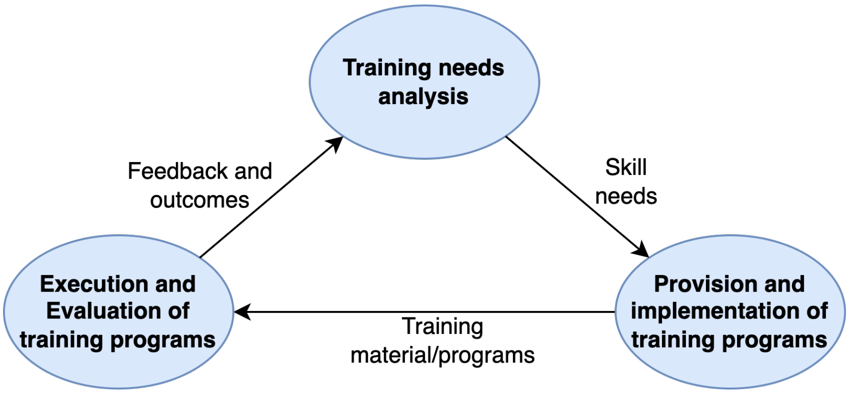

Training needs analysis
Skill needs
Provision and implementation of training programs
Training material/programs
Execution and Evaluation of training programs
Feedback and outcomes

Figure 2.1 – The training cycle, beginning with training needs analysis, and ending with the evaluation of training programs created or otherwise acquired on the basis of the analysis.

(Gould et al., 2004). As mentioned before, this can be done on the four levels of individual, department, company/organization, and profession (Chiu et al., 1999), although only the latter two are dealt with in this work, as the scope of the former two is too small for the top-level research question.

Training is a cyclic process, and it begins with training needs analysis. Once the analysis is performed and the training needs are identified, training programs pertaining to the identified needs are either designed or obtained from an external source. The final step is the evaluation of the training programs' impact on the target demographic (which can be employees in a company or a wider, profession-level audience) in order to see whether the programs achieved their goals (Gould et al., 2004). These steps can be seen in Fig. 2.1.

As mentioned earlier, training needs analysis deals with "gaps" between current skills and needed skills. This concept of a "skills gap" has been discussed at length in the literature, and the question of how real it is has been the subject of heated debate (Carter, 2014). Many corporations claim that they are unable to find suitable people to hire because of schools' failure to equip them with the right skills, while some argue that the problems lie in corporations' refusal to offer suitable wages and their lack of appropriate internal training (Cappelli, 2012). In either case, it is undeniable that today's labor market has been significantly impacted by disruptors such as AI, automation, and other technologies involved in Industry 4.0 (Schwab, 2017). These disruptors bring with them both process and product innovations that create significant skill gaps, as they bring new skills to the fore and render others outdated (Gopalakrishnan and Damanpour, 1997; Brynjolfsson and McAfee, 2014, 2011). In addition, during the course of the present dissertation, the COVID-19 pandemic swept across the world and significantly disrupted work in many industries, creating demand for new skills required for working from home, among others (Bai et al., 2020). All of this evidence points towards the importance of large-scale TNA: as educational organizations fail to keep up with the fast-changing landscape of skills, the average workers suffer (Brynjolfsson et al., 2018). The deprecation of their old skills and the inability of educational institutions to equip them with new ones result in worse economic outcomes for workers, while corporations reap the benefits of improved productivity and reduced labor costs brought about by automation (Brynjolfsson and McAfee, 2014). The impact of these disruptive factors is far and wide, and many professional domains have recognized the need to keep up (Lee and Mirchandani, 2010; Johnston, 2018). The large number of corporate reports on the subject, both by consulting firms and by educational platforms (Strack et al., 2020; BGT, 2019; Coursera, 2019) is another sign that this need for staying on top of the trends has been recognized. Therefore, it is imperative for educational institutions to keep up with the trends and enact curricular change in order to give workers the skills they need, and there are concrete disadvantages to *not* identifying these needs ahead of time.

Identifying skill needs and curricular change are relevant and necessary both for traditional educational institutions (e.g. universities, vocationals schools) and to institutions that support **lifelong learning** – the continued acquisition of new skills beyond the traditional school/university system (Laal and Salamati, 2012). Much like more traditional educational platforms, lifelong learning platforms as Massively Open Online Courses (MOOCs) have curricula (if in different formats) that will become outdated in the face of new skills if skill need identification is not prioritized. Lifelong learning is a particularly important learning paradigm in the context of the present dissertation, since the considerable disruptions brought about by the factors mentioned earlier result in a constant shifting of the skills landscape, breaking down the traditional dichotomy of the school as a place and time for learning versus the workplace as a place and time for applying what has been learned (Fischer, 2000). Such a change of paradigm requires not only the expansion of skill training and on-the-job learning, but also a change of mindsets among both the educators and the learners towards learning itself. Empowering public educational institutions in the face of such change is a tremendously important goal, given the fact that in this new economy, the gap between the work opportunities for the more educated and the less educated is widening and inequalities stand to intensify even further (Fischer, 2000; Field, 2000).

2.1.2 The methods of training needs analysis

A wide variety of methodologies have been developed for training needs analysis. The 2018 report by the International Labour Organization and the Organization for Economic Cooperation and Development (ILO and OECD, 2018) provides a thorough overview of most modern large-scale training needs analysis methodologies, many of which are used by governmental organizations. These methodologies are as follows:

- Surveys, interviews, focus group discussion, and workshops with domain experts
- Employer-employee skill surveys
- Gradute and student surveys
- Vacancy studies
- Quantitative forecasts at various granularities, using formal or predictive models

Surveys, interviews, and focus group discussions with domain experts are used to gather their opinions, thus leveraging their labor market knowledge. Methodologies involving the collection of information from experts make up a significant portion of the methodologies in the literature, especially in profession-level studies (Gould et al., 2004). **Surveys and interviews of employees and managers** are another highly popular methodology, allowing employees to report the skills that they believe they lack, and allowing managers and employers to report on the skills missing in their teams, based on the performance of their employees (Gould et al., 2004; Handel, 2012; Prinsley and Baranyai, 2015; Lee and Mirchandani, 2010). The advantage of such methods is that they approach the problem directly, leveraging the knowledge of people who are aware of market trends or of their own skill needs. This makes them particularly useful for company-level TNA (where responses can theoretically be elicited from *everyone*). However, especially at larger scales, they often suffer from high subjectivity and the representativeness of their respondents is a potential problem. This is due to the fact that their data is collected from a relatively small number of people, since their collection methods do not scale well. Among these methods, survey methods have the potential to scale better, but attaining high enough response rates can be difficult, especially if the survey's respondents are experts and executive-level managers (Baruch and Holtom, 2008; Fan and Yan, 2010). In addition, it has been shown that repeatedly surveying the same group of people can result in a considerable drop in response rates (Porter et al., 2004).

In educational institutions, **graduate and student surveys** are used (Fowler et al., 2014; Stevens et al., 2011). Graduate surveys inform the institution of the skills needed in the labor market that their programs do not equip students with. Student surveys, on the other hand, along with analyses of student performance data, can inform the institution of the gaps in the students' knowledge, and the shortcomings of their programs as perceived by the students themselves (Fowler et al., 2014). These methods essentially have the same advantages and disadvantages as the survey methods involving experts and employees, just in a different context.

Job vacancy studies are a type of labor market study that allows for the identification of skill gaps on the professional level (Hosen and Alfina, 2016; Dawson et al., 2019; Gallivan et al., 2004). These studies either look at job ads directly analyzing labor market demand for skills, or look at the positions that go unfilled for longer, thus analyzing the gap between what employers need and what potential employees have to offer. They can involve surveys of employers (especially when unfilled job positions are the focus), or job ad data alone. Since their focus is on analyzing the present state of the labor market, they are mostly useful for understanding short-term demand, and can suffer from subjectivity if employer surveys are involved (ILO and OECD, 2018). A closely related but more general family of methodologies is **quantitative forecasts**, which are projections of labor market demand for various skills in the short or long term. These are either based on formal economic models (Cambridge Economics, 2019) or on simpler non-formal predictive models. Such models use a variety of data sources, particularly including economic census data and past labor market demand for skills, at times combining them with expert input for fine-tuning (Cambridge Economics, 2019). These predictive methods provide indispensable insights into the future of the labor market, and suffer from less subjectivity than methods based on expert input (ILO and OECD, 2018). However, both the formal and statistical models suffer from limitations. A predictive model that does not formally model the underlying system (or is, in other words, not structural) can be vulnerable to the system's reactions[1]. This means that policy changes based on the model's predictions will affect the system, potentially changing its behavior and rendering the predictive model less accurate (as its data came from the system's old behavior, not the new) (Einav and Levin, 2014). Formal models, on the other hand, may fail to correctly model the underlying system, and are often quite data hungry. In addition, these models are usually much better at short-term prediction (often 1-3 years ahead) than longer-term predictions of 5 to 10 years ahead, as things get less predictable the further we go (ILO and OECD, 2018). In addition to these caveats, economic censuses and other such administratively curated data, which these models usually rely on, are only periodically collected because of their difficult and time-consuming collection (Horton and Tambe, 2015), meaning that such analyses can only be performed periodically.

All of these methodologies are used in a wide variety of professional domains (ILO and OECD, 2018). The Swiss system for profession-level TNA in vocational education uses committees of experts (who revise the training curricula every 5 years by default) combined with employer-employee surveys, and thus has the same shortcomings that such methods suffer from, being very time-consuming and thus largely limited to being conducted once every five years (SKBF and CSRE, 2018).

The relationships between the five types of data introduced in Chapter 1 and the types of methodologies discussed above are demonstrated in Fig. 2.2. Bear in mind that these data source types are not exhaustive and do not include such data sources as economic censuses or other curated data (e.g. rankings of companies such as Fortune 500), as these data sources were only used (or intended to be used) as auxiliary sources during the course of this dissertation,

[1] This is a problem that the present work will also have to deal with in later chapters, and it will be discussed in more detail there.

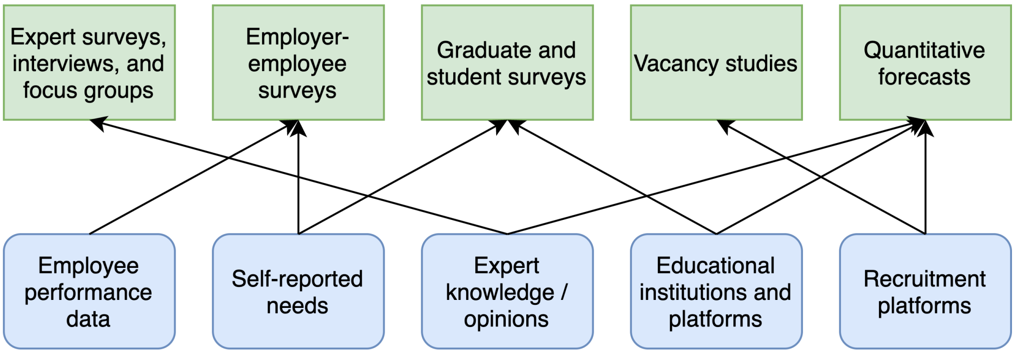

Expert surveys, interviews, and focus groups
Employer-employee surveys
Graduate and student surveys
Vacancy studies
Quantitative forecasts
Employee performance data
Self-reported needs
Expert knowledge / opinions
Educational institutions and platforms
Recruitment platforms

Figure 2.2 – The relationship between existing training needs analysis methodologies and the data sources introduced in chapter 1 as relevant to the problem tackled in this dissertation.

not as centerpieces.

2.1.3 Big data, skill needs, and lifelong learning

When it comes to identifying skill needs, most of the methodologies presented in the previous section are lacking both in agility and in granularity, and they often do not deal with latent profession-level training needs. As discussed previously, the rapid changes brought about by Industry 4.0 and COVID-19 require the quick adaptation of educational curricula, and the granularity of the new skills means that identifying granular skill needs as quickly as possible is a necessity. This is why a methodology based on online big data sources can be useful: online data sources (both educational and labor-related) provide granular data on skills and individuals, cover large scopes, and are always-on and thus do not require collection in the traditional sense (Horton and Tambe, 2015). This is stark contrast, for example, to surveys, which have to be collected, cannot be very granular at larger scales, and cause survey fatigue if sent repeatedly to a particular target audience (Baruch and Holtom, 2008). This is why the potential of Big Data for the identification of skill needs has been acknowledged not only by corporations, but by governments and international organizations as well (Strack et al., 2020; ILO and OECD, 2018; Haskel and Holt, 1999) (and the present dissertation itself is funded by a government body, further proving this point).

There are two main groups of big data sources that are relevant to the present dissertation: digital platforms used for education (regardless of whether or not they were designed for educational purposes), and online hiring platforms. There are some data sources that could be considered "big data" by virtue of their size (e.g. economic census data), but which are excluded here due to the fact that they lack at least one of the three desired qualities: large, pre-existing, and continuously-updated[2]. The data sources considered in this dissertation

[2] Such data sources *could* however be used as auxiliary sources to enrich others, and the later chapters will

can be of two kinds: some provide **a reflection of the skill needs of a population**, while others show the **response of educators to the skill needs of a population**. For example, online hiring platforms **reflect** how much demand a skill has among employers in a particular labor market, whereas general Q&A websites essentially **reflect** the skills that various individuals are trying to learn or use through the questions they ask. A prime example of the second type is MOOC platforms, where one can see what courses have been created **in response** to the need for various skills (although MOOC usage data can be considered to be of the first type: a reflection of how people interact with the MOOC and the skills they need and acquire). Both groups can be used for the purposes of the present work, even the ones that show the response of educators to skill needs. This is due to the fact that even though educators in *some* labor market may have identified *some* of the important skill needs, aggregating what they have identified and combining it with data from other sources can result in deeper insights into skill needs and the identification of more of them. Therefore, the rest of this section will deal with the existing literature on digital hiring platforms and digital platforms used for education, which may at times be rather sparse. The much more significant body of literature on the use of big data for lifelong learning will also be explored in parallel.

Platforms for or supporting digital education

Many different kinds of digital software and platforms (many of them online) have been used to support lifelong learning. These platforms can be classified along two axes:

1. Purpose-built educational software, versus software that was not designed for education from the get-go. The first group includes MOOC platforms such as Coursera, edX, Udacity, and Udemy (Conache et al., 2016), Q&A platforms such as Stack Overflow (Ishola and McCalla, 2016), peer help systems as PHelpS and iHelp (Vassileva et al., 2016), and personal knowledge managers with collaborative features such as Diigo (Field, 2000; Estelles et al., 2010). The second group includes online video sharing platforms such as YouTube, social networks such as Twitter and Facebook, blog platforms, and online videoconferencing tools which allow people to communicate - peer-to-peer and over long distances - for educational matters (Field, 2000).

2. Profession-scale educational platforms, versus those that are more limited in scope (e.g. to an organization). The scope of MOOC platforms and Q&A platforms depends on the size and diversity of their audience, but the platforms we have discussed before, such as Udemy, Coursera, and Stack Overflow are all profession-level in scale. Smaller-scale data, such as data from peer help systems, organization-specific Q&A platforms (e.g. Moodle forums for a university), and individual MOOC data can be used in company-level

briefly touch upon some examples.

studies.

Among the data sources described above, many have sensitive data (e.g. instant messaging, company-specific forums and MOOCs), making them hard to acquire, especially for profession-level studies. Others, such as Q&A platforms and public MOOC providers, are promising sources whose data is often available online and can be either scraped or directly downloaded. Therefore, a deeper dive into the literature around these two types of platforms is necessary.

Q&A websites have had a huge rise in popularity in recent years. The Stack Exchange family is a prime example of this, with its most famous member being Stack Overflow, which is a Q&A platform for software developers. These platforms have moved away from simply providing good answers for the question askers, and towards becoming repositories of community-curated knowledge (Anderson et al., 2012). A considerable body of research exists on these websites , especially on Stack Overflow. These papers tackle subjects such as the content of questions and their trends(Barua et al., 2014; Allamanis and Sutton, 2013), the interactions of Stack Overflow with other platforms such as GitHub (Vasilescu et al., 2013), and many others. Stack Overflow's popularity and dynamics, such as the high level of moderation present on the website(Correa and Sureka, 2014; Ponzanelli et al., 2014), the quick response rates (Barua et al., 2014), and the quality control measures make it an invaluable data source when analyzing the software industry. This, coupled with the growing popularity of the Stack Exchange model, makes Stack Overflow a valuable data source for methodologies that identify present and future skill needs. Despite the significant number of studies on Stack Overflow, only a few of them present methods for analyzing the topics and skills discussed there and their trends (Barua et al., 2014; Yang et al., 2016; Ishola and McCalla, 2016). Some of these (Barua et al., 2014; Yang et al., 2016) track the popularity of various topics over time, while others (Ishola and McCalla, 2016) track the knowledge needs of individual learners on Stack Overflow. Overall, these platforms represent an untapped potential for the analysis of skill needs. The steady expansion of the Stack Exchange family of websites to professional domains other than software programming means that any methodology developed for Stack Overflow may soon become applicable to many other domains as well.

MOOCs are another novel type of digital education, and they too have experienced a quick rise to prominence in recent years, owing to their usefulness as an instrument of lifelong learning. Much scholarly work has been conducted on these platforms (Ebben and Murphy, 2014; Conache et al., 2016; Zhu et al., 2018; Bozkurt et al., 2016). Some MOOC platforms, such as Coursera, tend to mimic actual universities: they have academic schedules, with exercises, quizzes and exams, and the possibility of earning certificates for single courses or for entire programs. On the other end of the spectrum are MOOC platforms that offer individual skill-based courses, such as Udemy, with self-paced and skill-centric MOOCs. Udemy is of particular note because its business model is one where any person can sign up to become a content creator and create (free or paid) MOOCs for others to use (Conache et al., 2016); this is in contrast to platforms such as Coursera, Udacity and edX, where the content providers are universities, organizations or corporations(Conache et al., 2016). Some work has been done on skill trends in MOOCs, mostly by the MOOC platforms themselves, such as Coursera (Coursera, 2019) and Udemy (Wai, 2016). These studies use enrollment data to understand the demand trends for different skills in different areas of the world. Most existing MOOC research, however, focuses on the students' experience, their motivation, their retention, and on the design and assessment aspects of the MOOC (Zhu et al., 2018; Bozkurt et al., 2016). Again, the potential of MOOCs for analyzing the current state of the labor market and for analyzing skill needs is as of yet quite untapped.

Digital hiring platforms

When it comes to understanding labor market skill trends, analyses of massive job ad collections a popular choice, as they manifest the demand for skills on the labor market. Such analyses are conducted by governments (ILO and OECD, 2018), by academics (Gallivan et al., 2004; Lee and Mirchandani, 2010; Matsuda et al., 2019; Gurcan and Cagiltay, 2019), and by corporations that collect or host this type of data (Strack et al., 2020; LinkedIn, 2019). The latter is particularly interesting, as companies like LinkedIn and Burning Glass Technologies provide labor market insights on skills (as well as on job titles, companies, etc.) as part of their premium services. Previous labor market research on skills using these data sources often either focus on the *currently* in-demand skills (Hiranrat and Harncharnchai, 2018; Papoutsoglou et al., 2017), or on higher-level skill or job trends (Gallivan et al., 2004; Lee and Mirchandani, 2010; Matsuda et al., 2019; Gurcan and Cagiltay, 2019). Therefore, such data has untapped potential for skill need analysis and curricular change.

The study of skill trends using job ad data presents interesting opportunities. Curricular change involves two aspects: adding up-and-coming skills that are rising to prominence, and gradually phasing out the skills that become obsolete. Job ad data is able to show both (Strack et al., 2020), and can help distinguish between the declining skills, the already-popular skills that are growing in popularity, and the skills that are rising from relative obscurity (Strack et al., 2020). These concepts have served as inspiration in the later studies of the present dissertation.

2.2 Research overview

The final part of this chapter deals with the research path taken in this dissertation, and takes a more chronological approach, since it deals with the development of the methodology that eventually became the main contribution of the present work. This research path consists of

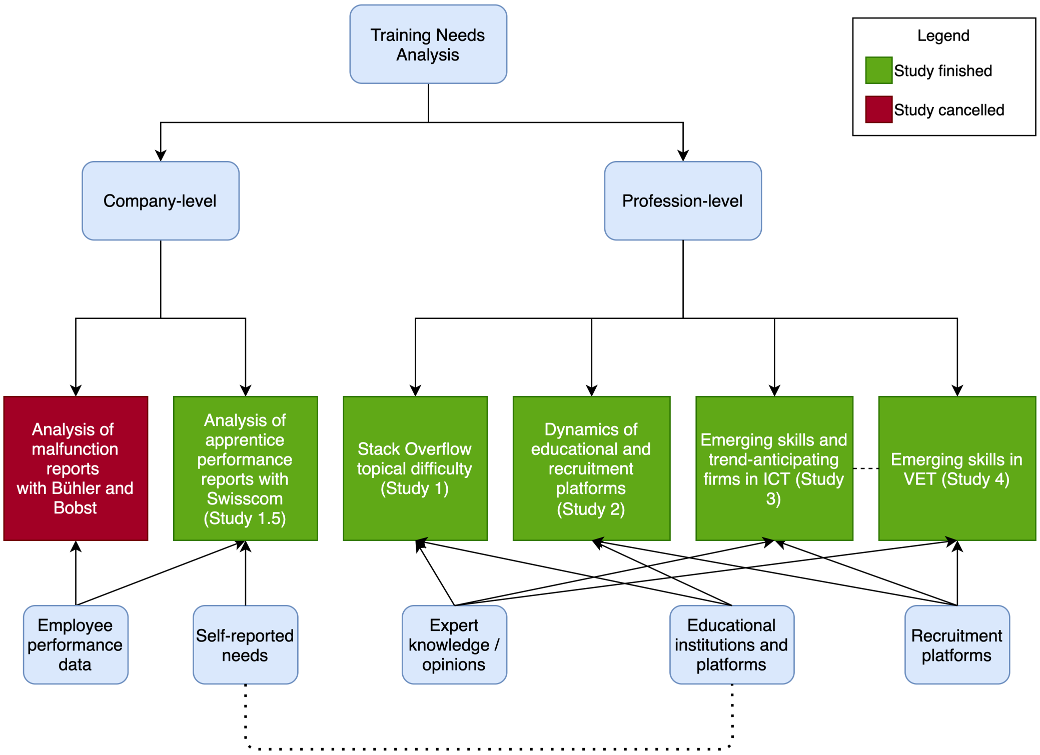

Training Needs Analysis
Legend
🟩 Study finished
🟥 Study cancelled
Company-level
Profession-level
Analysis of malfunction reports with Bühler and Bobst
Analysis of apprentice performance reports with Swisscom (Study 1.5)
Stack Overflow topical difficulty (Study 1)
Dynamics of educational and recruitment platforms (Study 2)
Emerging skills and trend-anticipating firms in ICT (Study 3)
Emerging skills in VET (Study 4)
Employee performance data
Self-reported needs
Expert knowledge / opinions
Educational institutions and platforms
Recruitment platforms

Figure 2.3 – The two levels of TNA, along with the approaches tried in this work and the data source types used in each of them.

several studies which can be seen in Fig. 2.3, all of which aimed to answer the top-level research question. The success or failure of each approach informed the design of the next. This section will also discuss some of the more minor studies that were conducted but abandoned, and the subsequent chapters will deal exclusively with the major studies (studies 1 through 4) and their results.

Two central ideas will come up repeatedly in the following discussions. The first is the theory of "diffusion of innovations", which describes the process through which **innovations** get **communicated** among the members of a **social system** over **time** (Rogers, 2010). The special case that applies to the present dissertation is the process through which newly appearing skills (which are **process innovations** (Gopalakrishnan and Damanpour, 1997)) are, over time, adopted by different actors (companies and other organizations) in a professional domain or industry. The theory posits that this process generally follows an S-curve, in which "innovators" are the first to create and adopt the innovation, and then once a critical mass is achieved, the innovation is adopted by so-called "early adopters". Afterwards, the adoption accelerates and the innovation is diffused into the "majority" of actors until the pool of remaining potential adopters gets smaller, and the innovation continues to spread, more slowly, to the "laggards"

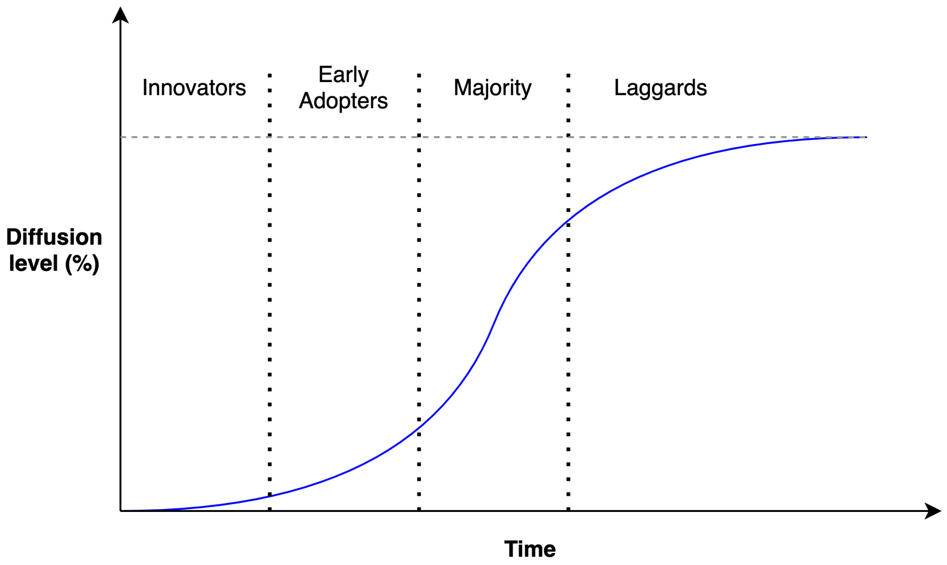


Figure 2.4 – The characteristic S-curve of innovation diffusion. The vertical dotted lines delineate the groups of adopters based on how early they adopt the innovation, and the horizontal dashed line shows the limit that the curve approaches as everyone “adopts” the innovation. In practice, of course, the shape is not always an ideal S, and the limit can be lower than 100% diffusion. This dissertation will only use the general idea of this theory, rather than the specifics.

(Rogers, 2010). This process is illustrated in Fig. 2.4. Although this description of the diffusion of an innovation is not universally applicable and the S-curve is not necessarily correct (Rogers, 2010), what matters in this dissertation is the early part of the curve: innovators, early adopters, and the breakout of the innovation into the majority. Since the main goal is to provide educators and training providers with *early* warning about important skills that will be in need of training programs, the early adopters are much more valuable than the “majority” adopters.

The second idea, building on the first, is that these early adopters - be they experts, managers, practitioners, or learners - leave online traces of their “adoption” of new skills. Such traces can come from their learning activities (e.g. question on Q&A websites, signing up for MOOCs or creating MOOCs), their networking activites (e.g. connecting with others on LinkedIn), their recuitment (e.g. posting job ads), et cetera. The researcher’s task, then, is to identify the platforms where these traces can be found, understand their context and what they mean, and aggregate them to learn about the skills landscape of the domain of study. The studies conducted as part of this dissertation can all be characterized as different attempts to perform this task, to varying degrees of success.

2.2.1 Study 1: Topical Difficulty on Stack Overflow

Building on the idea that the Q&A platform Stack Overflow is one of the fastest platforms when it comes to manifesting the needs of learners in the software programming domain, a decision was made to use Stack Overflow as the basis of the first study. **Study 1** was designed on the concept of the "difficulty" of learning a topic, using Stack Overflow question response times to estimate such difficulty for various topics. Then, the input of practitioners was used to verify this concept by comparing the estimated difficulties of different topics. The goal of this study was to create a "difficulty ranking" between topics, which then could be used by training providers, who have a set of topics to teach based on their target audience and organizational priorities, to prioritize their course creation efforts towards the more difficult topics among those. This approach was unsuccessful, as the concept of topical difficulty was found to be too ambiguous, and practitioners tended to disagree about the relative difficulties of topics. Therefore, rather than delving deeper into that direction, a change of course was chosen for the second study.

2.2.2 Study 2: Dynamics of Online Hiring and Learning Platforms

Study 2 investigates the dynamics of the appearance of *new* topics on online learning and hiring platforms, using Stack Overflow (Q&A), Udemy (MOOCs), and Stack Overflow Jobs (job ads) as the data sources. The hypothesis was that most topics would appear first on Stack Overflow due to the fact that the creation of its data (i.e. questions) was more driven by individuals, and often by novices. The ultimate goal was to predict the appearance of new topics in MOOCs or job ads based on signals from Stack Overflow (e.g. question and vote counts). If successful, such a prediction would allow the provision of early warning to training providers about the new topics that were showing signs of increasing importance. The results proved the hypothesis, showing that platforms where content creation was more by novices than by experts, and more by individuals than by groups were faster at manifesting new topics, although the prediction task did not yield useful results. This was due to a scarcity of sufficiently strong signals on Stack Overflow that would discriminate between those new topics that later appeared in MOOCs and job ads, and those that did not.

The second study showed that the concepts of "new topics" and "first appearances" were not practical for solving the fundamental question of the present thesis (at least in the software programming domain), and that a focus on the rise of topics to importance (rather than their first appearance) would be more useful. In addition, up to this point, Stack Overflow had been the main data source of this work due to a belief in its potential for the *very* early identification of important topics. However, for the third study, job ads were chosen instead. On one hand, job ads would allow for much greater generalizability, as they are available for any professional domain, and the hiring demand for a skill is closely related to the need for training in that skill. On the other hand, the relatively short delay that Study 2 found between the appearance of new topics on Stack Overflow versus job ads (which was usually around half a year), combined with the realization that the earlier expectations of early identification may have been overly ambitious, meant that job ads would not necessarily be too "slow" for the task of early identification.

2.2.3 Study 3: Emerging Skills in ICT

This led to the concept of "emerging skills" becoming the centerpiece of **Study 3**, defined as skills with previously low hiring demand that have recently experienced a surge in hiring demand based on job ads. The goal of Study 3 was to predict the "emergence" of skills in the near future using signals from one year of past job ad data, thus giving early warning of the increasing importance of these skills to training providers. This classification approach was successful, with the emerging skill classifiers beating several baselines and therefore showing the practical usefulness of the concept of emerging skills. This proved that past job ads do contain signals that can help predict a future rise (or fall) in importance, and that a relatively simple classification model with no knowledge of the relationships between skills or the bigger market trends can already achieve good performance in short term growth projection. Furthermore, the important signals were identified, and the limitations of the classifier (including where it fails and how well it predicts further into the future) were quantified.

2.2.4 Study 4: Emerging Skills in VET

Building on the success of the third study, **Study 4** was dedicated to the question of generalizability and to tying the work back to the VET domains: is the concept of emerging skills only useful in a rapidly evolving domain like software programming, or is it also possible to predict the emergence or non-emergence of skills in VET domains that evolve more slowly? For this study, two professional domains falling under the VET umbrella were chosen: logistics, and healthcare. These two domains were found to be very different from ICT in terms of the way emerging skills behave, and the prediction of emerging skills was generally not successful. However, the analysis of this failure resulted in a better understanding of the different dynamics of these domains, and allowed for the development of measures for checking the feasibility of such a prediction.

2.2.5 Other studies

As shown in Fig. 2.3, two company-level approaches were also tried as side projects. As previously mentioned, acquiring enough company-level data to aggregate them into profession-level insights can be prohibitively difficult. This is because companies are often very reluctant to share performance or training need data on their employees with third parties due to privacy concerns, and non-disclosure agreements with each company could render the multi-company aggregation legally questionable. Despite all of these hurdles, a few individual company-level studies were tried. Most of these failed before they had begun, as the companies ultimately refused to share their data (the attempted study with Bobst and Buhler, shown in red in Fig. 2.3 is an example). However, one case, which was a study on apprentice performance reports with Swisscom, was relatively successful. Unfortunately, since the focus of the performance reports was mostly on the soft skills and the work ethic of the apprentices, the data was unsuited for training needs analysis, and a decision was therefore made to focus on larger-scale, profession-level datasets for the subsequent studies.

3 Study 1: Topical Difficulties on Stack Overflow

3.1 Introduction

In the previous chapter, two ideas were discussed. The first was the concept of “diffusion of innovations”: innovative technologies and skills that achieve widespread adoption diffuse among practitioners in a process whose curve is S-shaped, and initially involves a slow spread among early adopters. The goal of the present work is ultimately to detect this diffusion process as early as possible. The second idea was that the online learning traces left by learners, practitioners, and experts alike can be leveraged to understand the skills landscape of a professional domain. In particular, aggregating the educational activities of *learners* on online platforms can paint a picture of their training needs. A concept that can help with this aggregation is the “difficulty” of topics, i.e. the difficulty involved in learning a skill or learning how to use a technology. If the relative difficulties of topics can be ranked, then a difficulty-ranked list of topics can be given as the summary of the learners’ training needs: for topics that are seen as important to learn, the higher the difficulty, the greater the need for suitable training materials.

However, topical difficulty is not an easy topic to tackle, for two reasons. First of all, topical difficulty means different things in different contexts, thus necessitating a more precise definition. For example, in an Information Retrieval (IR) context, the difficulty of a topic refers to the difficulty of retrieving the correct results for queries that include the topic (Gienapp et al., 2020). In an educational context (which is the type of context relevant to this thesis), the concept of difficulty refers either to the “perceived” difficulty of a topic by learners, which shows how “afraid” they are of the topic (Okebukola and Jegede, 1989; Bernstein et al., 2013; Hall et al., 2018; Scott and Fensham, 1977), or to the amount of struggle involved in learning the topic, such as the time needed to learn it or the performance of students when their knowledge of that topic is tested (compared to their performance on other topics) (Ehara et al., 2012). The second issue with the concept of topical difficulty is that it is subjective and dependent on the learner, particularly when it comes to perceived difficulty, which is self-reported.

Despite the ambiguity and subjectivity of the concept of topical difficulty, it presents an interesting research opportunity when combined with online community Q&A platforms. Such platforms present both the questions being asked by learners and the answers being given by experts on a very large scale and in a variety of topics. Developing an approach for estimating topical difficulties using such data would allow experts, whose target audience require a particular set of skills, to prioritize their attention and effort towards the more difficult skills in that set (while taking their information about their target audience and other factors into consideration). One such platform - which has already been discussed in the previous two chapters - is Stack Overflow, a collaborative questions and answers (Q&A) website focused on questions related to the software programming domain. Users can post new questions, answer existing ones, and they can upvote or downvote a post (both questions and answers). The poster of a question can choose one of the answers as the “accepted answer”, designating that response as the most relevant or most helpful one. A variety of topics are discussed on Stack Overflow, and the questions are also tagged (with up to 5 tags per question) in order to indicate their topics, such as the relevant programming languages, frameworks and packages (although questions are not always tagged optimally, particularly due to the 5 tag limit). Stack Overflow has grown to be one of the largest Q&A communities in existence and an important reference point for programmers, with very short question response times (which is the time between the posting of a question and the posting of its accepted answer) (Hanrahan et al., 2012; Wang et al., 2018). There is a large body of existing work on Stack Overflow, investigating the structure and inner workings of the community (Correa and Sureka, 2014; Anderson et al., 2012), the topics discussed and different types of questions asked, et cetera (Johri and Bansal, 2018; Barua et al., 2014; Ponzanelli et al., 2014; Allamanis and Sutton, 2013). The previous work on Stack Overflow also touches upon topical difficulties, using measures such as aggregated question response times, the percentage of unanswered questions, or average number of responses per question to characterise the difficulty of topics (Yang et al., 2016; Hanrahan et al., 2012). However, the previous work on the subject generally lacks an educational focus: it does not seek to validate the correspondence of this estimated difficulty to educational definitions of difficulty, and does not deal with the use of such measures for training needs analysis.

The goal of this study is to develop a methodology for estimating the relative difficulty of topics discussed on Stack Overflow, and to investigate the relationship between this measure and the perceived difficulty of those topics by practitioners in the software programming domain. After topic extraction from Stack Overflow questions, the difficulty of those topics is estimated using a methodology that predicts the response times of questions associated with each topic, taking into account the confounding factors that can affect response times. Using this measure, relative difficulties are established between pairs of topics. Some of these pairs are then sampled and compiled into a randomized survey and presented to software developers. For every pair of topics, they give their opinion on which one is a more difficult topic, and the aggregated opinions are then used to evaluate the difficulty measure. In addition, the respondents are asked to give a free-form (but guided) explanation of their opinion for each question, which can be used to gain insights into the “why” of their responses and to compile a list of difficulty “types”, i.e. what makes a topic more difficult than another. These insights can then be used to refine the difficulty measure and to understand what it exactly measures when it comes to perceived difficulty.

The results of this study were mixed. On one hand, the predictive methodology designed to measure the effect of topics on response times found statistically significant effects. This means that even after taking a variety of confounding factors into account, the topic of a question does affect its response time and some topics receive responses significantly faster than others. On the other hand, the results of the survey found a rather weak relationship between the estimated relative difficulties and the developers’ perceptions of relative difficulty, and some users used similar reasoning (as given in their free-form responses) to give contradictory responses about the same pair of topics. In addition, the topics were found by some respondents to be ambiguous. As a consequence of these results, it was decided that further work towards understanding topical difficulties and refining the methodology would yield diminishing returns, and that a change of direction was necessary.

In the next sections, a review of previous work will be given on both topical difficulties and on Stack Overflow. Afterwards, the predictive methodology developed for this study will be discussed, and the structure of the software developer survey will be laid out. Then, the results of both the predictive methodology and the developer survey will be presented, and their implications will be discussed. Finally, a summary will be given of the contributions of the study and the reasons for moving on.

3.2 Related work

3.2.1 Topical difficulty

The use of topical difficulty was introduced earlier as a potential way of conducting training needs analysis. However, topical difficulty itself is a concept that needs to be specified, as there are many different types of difficulty, and its disambiguation is a necessary step before its measurement.

In general, when it comes to the difficulty of tasks, a distinction is made between perceived, physiological, and performance-based difficulty (Borg et al., 1971), although most of the works on mental tasks focus only on perceived and/or performance-based difficulty (Wall and Knapp, 2014). Perceived difficulty has to do with how learners or practitioners feel about a task, physiological difficulty is concerned with the toll the task takes on the body, and performance-based difficulty is based on how well the task is performed on average. The idea of using performance as an “objective” measure of difficulty has long been phased out, since there are subjective costs involved in attaining said performance (Borg et al., 1971), and perceived difficulty has been shown to affect performance in its own right (Okebukola and Jegede, 1989; Newton and McCunn, 2015). Given the fact that lifelong learning is an integral part of the workers’ lives, it is important for any high-level training needs analysis method to take such subjective costs into account, making perceived difficulty the most appropriate measure for the present work.

Previous work has established the theoretical basis of perceived difficulty and proven its measurability, which is generally deemed to be harder for mental tasks than for physical ones (Borg et al., 1971). A variety of methods have been used for measuring perceived difficulty: the most common method is using questionnaires (Borg et al., 1971; Hall et al., 2018), while biometric sensors (Müller, 2015; Fritz et al., 2014) and eye-tracking (Whitehill et al., 2008) have also been used in more recent work. What is of particular interest in this study, however, is not the method of measurement, but rather what is being measured. Works such as (Müller, 2015; Fritz et al., 2014) focus on **task difficulty** in software development, i.e. how identifying the points of struggle for software developers can help develop tools to make sure they introduce fewer bugs in their code. Others works, such as (Hall et al., 2018; Wall and Knapp, 2014; Scott and Fensham, 1977; Whitehill et al., 2008) focus on **learning difficulty**, i.e. the trouble students have when being taught certain subjects and developing ways of helping them learn better. In all cases, these methods seek to establish the relative difficulties of the topics of interest, thus creating a partial ordering that allows educators to focus on the most difficult topics (Hall et al., 2018). However, previous work often focuses on scales smaller than a profession, which is the scale of interest in this study.

3.2.2 Online Q&A: Stack Overflow

As introduced earlier, Stack Overflow is an immensely popular Q&A platform where software developers can ask and answer questions about a variety of questions related to software development. The Stack Overflow community is very quick at answering questions, with around 63% of questions answered less than 1 hour after being posted, and less than 10% unanswered after the first day (Bhat et al., 2014). This quickness implies that many of the questions asked on Stack Overflow are on topics that are widely known, and are *easy* to answer for the community. In other words, there are plenty of experts on the website for the topics of those questions. On the other hand, there are questions that take a lot longer to get an answer, or even remain unanswered. This means that the community has a harder time answering those questions, which could be due to a the smaller size of the expert community, the actually greater difficulty of answering those questions, or a combination of both. By looking at previous literature, it is possible to identify three potential measures for the difficulty of a topic (Bhat et al., 2014; Wang et al., 2018; Saha et al., 2013; Anderson et al., 2012):

1. The number of answers for each question of that topic
2. The proportion of unanswered questions of that topic
3. The response times of the questions of that topic

However, these measures cannot be directly attributed to difficulty, since many confounding factors affect these quantities. For example, (Bhat et al., 2014) design a pipeline that predicts whether a question is going to be answered very early or very late based on the features of the person posting the question, the person answering the question, and the question itself (although the topics of the question are not directly included in their model). They find that the most important features are those pertaining to the person posting the question and the person answering it, although many features (including many of the question's features) have a significant effect (Wang et al., 2018; Bhat et al., 2014). The importance of these confounding factors means that a more in-depth analysis is needed on the relevance of the aforementioned measures to actual topical difficulty, and such an analysis is missing in the literature. In addition, previous work does not engage with the question of connecting these *presumed* measures of difficulty with perceived or performance-based difficulty.

3.3 Methodology

Topical difficulty is, as discussed before, a useful concept for the analysis of training needs. Given a ranked list of topics that sorts them from easiest to most difficult, training providers can prioritize the more difficult ones for their training programs while taking any other important factors into account (e.g. as the priorities of their institution and their knowledge of their target audience). However, to ground this relatively vague concept, several steps are necessary. First of all the two parts of the phrase “topical difficulty” need to be clarified by specifying how the topics are identified and how their difficulty is estimated. This is what the next section will deal with. Afterwards, it is necessary to make the link between the estimated topical difficulty and “actual” topical difficulty. Due to the fact that the topics on Stack Overflow are incredibly varied and acquiring developer performance data on them is difficult, the type of difficulty that has been chosen for this study is **perceived** difficulty. In other words, this study posits that the estimated difficulty of a topic should correspond to how difficult practitioners perceive it to be. This step will use a developer survey that is distributed to software development practitioners in order for them to give their opinions on the relative difficulties of topics, which will then be compared to the estimated relative difficulties. The section after the next will deal with this survey.

The two fundamental underlying assumptions in this methodology are that:

1. It is possible to estimate the relative difficulties of topics in a data-driven fashion, and this estimated difficulty will, to some degree, correspond to the perceived difficulty of those topics.
2. Some degree of consensus exists in perceived difficulty. In other words, practitioners will agree, on some level, that certain topics (i.e. skills) are harder to learn and use than others, and that certain characteristics contribute to this relative difficulty.

This study investigates the degree to which each of these two assumptions is true.

3.3.1 Topical difficulty methodology

Identifying topics

Generally speaking, the topics that matter in this study are those that pertain to skills and technologies that can be learned. There are a variety of ways to identify topics in Stack Overflow questions: manual labelling, using existing Stack Overflow tags, and using topic modelling methods to extract topics in a data-driven fashion. Given the large number of topics and questions involved in this study, manual labelling is infeasible, leaving the two other methods as options.

Stack Overflow tags are user-created phrases that indicate the topics of Stack Overflow questions at various degrees of granularity. Each question can have up to five tags, which can be added to the question either by the user posting the question or by a moderator (Anderson et al., 2012). Users with a high enough reputation can create new tags when posting a question, and tens of new tags are created weekly. In addition, tags are monitored and edited by moderators, adding further to their quality. This makes them a useful tool for tracking fine-grained new topics on Stack Overflow. However, each question can have no more than 5 tags, and imperfect tagging by users means that quite often, questions have even fewer than 5 tags. In addition, the stated purpose of Stack Overflow tags is to help locate questions by describing their content, not to describe their context or nature, and tags that are more relevant to the latter two than to the former are deleted by moderators over time (and generally discouraged). These two factors limit the usefulness of Stack Overflow tags for the purposes of this study.

Topic modelling methods such as Latent Dirichlet Allocation (LDA), which are a staple in Natural Language Processing (NLP) tasks (Blei et al., 2003), are the other possible choice for this study. A topic model represents each document (which would be a question in the case of Stack Overflow) as a distribution over a number of topics, each of which is a distribution over words. The number of topics is usually set as a hyperparameter, and the model learns the distributions for the topics and for the documents from the documents that it is given (Blei et al., 2003). A particularly useful topic model for Stack Overflow is the correlated topic model (CTM), which, unlike LDA, can accommodate the correlations that could exist among topics (Blei and Lafferty, 2006). This is particularly important for Stack Overflow, since there is a lot of affinity between the various topics discussed on Stack Overflow, and assuming that they are uncorrelated can reduce the quality of the extracted topics. The "quality" of topics is measured through two metrics: the **log-likelihood of a holdout set** of questions under the topic (which measures how well the topic model explains unseen documents coming from the same data), and the **exclusivity** of words in the model, which measures how exclusive each word is to a particular topic. These quality metrics can be used in model selection in order to choose the best topic model type (i.e. CTM vs. LDA) and the best number of topics.

The advantage that topic models have over tags is that since they are extracted in a data-driven fashion, a question can be a mixture of as many topics as necessary, and there is no limitation on *what* the topics are. The main issues of using topic models are that topics need to be interpreted (since each is a distribution over words, unlike tags which are represented by a single word), and that some topics may capture linguistic features that are irrelevant to the semantic content of questions. Despite these challenges, topic models remain preferable for this study, since the limitations of tags are harder to address. The subsequent sections will discuss how the limitations of topic models were dealt with in this study.

Estimating difficulties

The related work section introduced three potential measures for topical difficulty, which will be reiterated here:

1. # of responses per question in topic
2. % of unanswered questions in topic
3. Response times of questions in topic

Although the ideal measure for difficulty would combine all three, this study uses the third measure. The first measure (# of responses per question) is closely related not only to question difficulty, but also to the number of ways there are for solving problems in a particular programming language, and is therefore less directly relevant to topical difficulty than the other

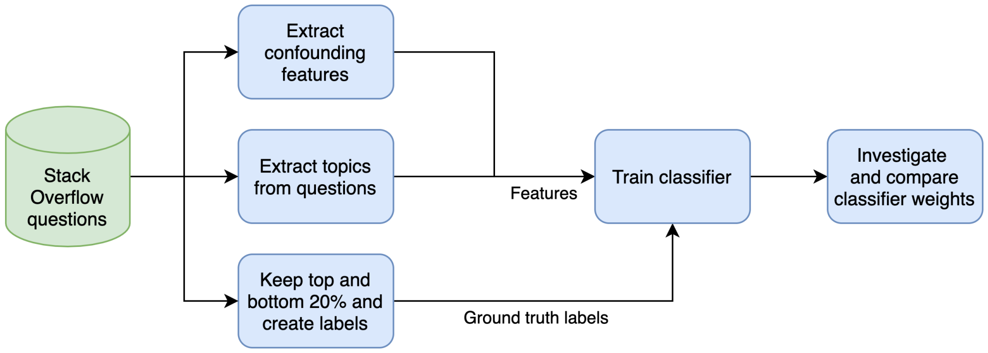

Stack Overflow questions
Extract confounding features
Extract topics from questions
Keep top and bottom 20% and create labels
Features
Ground truth labels
Train classifier
Investigate and compare classifier weights

Figure 3.1 – The pipeline for predicting response time class based on question topics and confounding features.

two. The second and third measures can both be useful, but only looking at unanswered questions makes the approach more vulnerable to cases where the user who posted the question neglected to mark an answer as accepted or where the question was badly formed and did not receive an acceptable answer. Therefore, for this study, the third measure will be used.

As discussed before, question response times are greatly affected by factors other than the topics of the question itself. Therefore, the first step in this study is to investigate the effect of topics on question response times, check if the effects are significant, and make sure that there are significant *differences* between the effects of different topics. In order to do so, a pipeline similar to the pipeline in (Bhat et al., 2014; Wang et al., 2018) is used, which can be seen in Fig. 3.1. A logistic regression classifier is used to predict, for each question, whether it has been answered very early or very late using a variety of features. Two types of features are used in this model: topics, and confounding features. There are a total of 46 confounding features: 16 features for the question and answer, and 30 features for the inquirer and responder. These features are those used in works such as (Bhat et al., 2014; Wang et al., 2018), with the exclusion of the tag-related features (as such features are already included through the extracted topics) and with the addition of the following features:

- The time of day when the question was posted (3 features, dividing the day up into 3 parts), in order to capture potential night/day differences
- Whether or not the question was posted on a weekend, to capture potential weekday/weekend effects
- The month the question was posted in, to capture the potential effects of holidays (e.g. Christmas)
- Code-to-length ratios for both the question and the answer

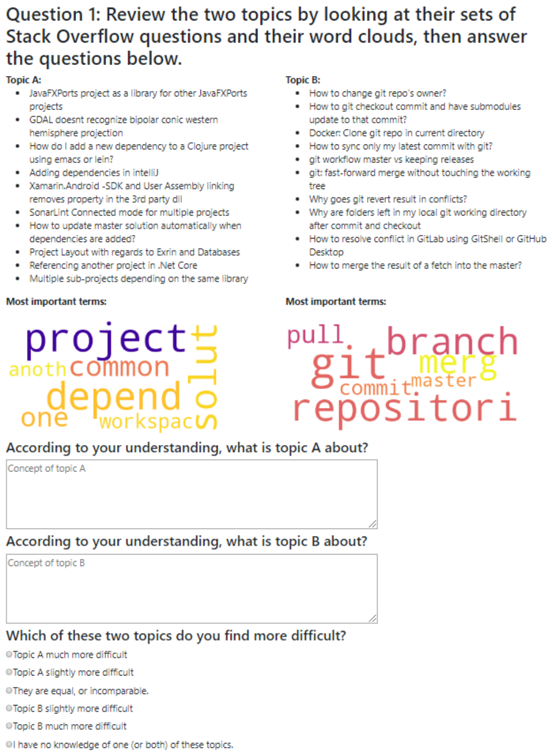
- The popularity of the question's topics in the month m when the question q was posted, defined as $\sum_{t \in topics} P_{q,t} P_{m,t}$, where $P_{q,t}$ is the proportion of topic t in question q, and $P_{m,t}$ is the overall popularity of topic t in month m. This is a measure of popularity that is "topic-agnostic", i.e. it aggregates the topic away and can capture the effects of popularity that are independent of the question's specific topics.

A full list of the features can be found in Appendix A. After the confounding features are computed, F tests are used on the polynomial degrees of each feature (up to a maximum of 4) in order to find the ones that have a significant univariate effect on the output, and those are kept while the non-significant ones are discarded. Once the full classifier has been trained, investigating its weights for each feature makes it possible to understand whether or not the effect of each topic is statistically significant (using Wald's t-tests on individual coefficients), and to compare the effects of various topics (using pairwise Wald's t-tests between pairs of coefficients). The comparisons between topics will induce a partial ordering on them. In other words, by taking the pairs of topics where the first topic has a significantly smaller contribution to late responses than the second topic, a "less difficult than" relation will be formed (which will be denoted by <), and taking all of these relations together can create chains such as (JavaScript < Node.js < TypeScript).

3.3.2 Developer survey design

The topical difficulty estimation pipeline produces, as its output, pairs of topics where one topic's effect on late question response times is significantly smaller than the other's. This relation is interpreted as the first topic being *less difficult to learn* than the other. In order to make sure that this interpretation is reasonable, it is necessary to verify the link between estimated and perceived difficulty by involving real developers who deal with those topics as part of their work.

To do so, two sets of *topic pairs* were created: those where the difference in estimated difficulty is significant, and those where it is not. Then, 35 pairs were sampled from the former set, and 35 from the latter. This formed a base set of 70 questions (not to be confused with Stack Overflow questions which were discussed previously), which the randomized developer survey would be built out of. The non-significant set is sampled from because it is possible that the difficulty estimation methodology would fail to recognize certain relative difficulty relationships. Only by comparing the developer input on the significant pairs versus the non-significant ones would it be possible to figure out if this methodology is succeeding at the task it is designed to do, which is to tell apart the topic pairs that have different difficulties from those that do not. With the main 70 questions sampled, and in order to be able to properly screen the respondents, a set of 5 screening questions were created, consisting of synthetic pairs of topics



Question 1: Review the two topics by looking at their sets of Stack Overflow questions and their word clouds, then answer the questions below.

Topic A:

- JavaFXPorts project as a library for other JavaFXPorts projects
- GDAL doesnt recognize bipolar conic western hemisphere projection
- How do I add a new dependency to a Clojure project using emacs or lein?
- Adding dependencies in intelliJ
- Xamarin.Android -SDK and User Assembly linking removes property in the 3rd party dll
- SonarLint Connected mode for multiple projects
- How to update master solution automatically when dependencies are added?
- Project Layout with regards to Exrin and Databases
- Referencing another project in .Net Core
- Multiple sub-projects depending on the same library

Most important terms:

Topic B:

- How to change git repo's owner?
- How to git checkout commit and have submodules update to that commit?
- Docker: Clone git repo in current directory
- How to sync only my latest commit with git?
- git workflow master vs keeping releases
- git: fast-forward merge without touching the working tree
- Why goes git revert result in conflicts?
- Why are folders left in my local git working directory after commit and checkout
- How to resolve conflict in GitLab using GitShell or GitHub Desktop
- How to merge the result of a fetch into the master?

Most important terms:

According to your understanding, what is topic A about?

Concept of topic A

According to your understanding, what is topic B about?

Concept of topic B

Which of these two topics do you find more difficult?

○ Topic A much more difficult
○ Topic A slightly more difficult
○ They are equal, or incomparable.
○ Topic B slightly more difficult
○ Topic B much more difficult
○ I have no knowledge of one (or both) of these topics.

Figure 3.2 – A screenshot of one question in the randomized developer survey.

in which one was clearly more difficult than the other. These questions therefore had correct and incorrect responses, which is something that generally cannot be said about the other questions in the survey, since perceived difficulty is a subjective concept. This brought the total number of questions up to 75.

The developer survey begins with a description of the task, asking respondents to answer each question based on which of the two topics they found more difficult to learn and master. It then presents 10 questions, randomly sampled out of the full set of questions with 7 "real" questions and 3 screening questions, to the respondent. An example of a question from the developer survey can be seen in Fig. 3.2. In each question, a pair of topics A and B are shown to the respondent, who then indicates the one they found more difficult to learn than the other. The topic is represented through two things:

- A word cloud, created using the distribution of that topic on words. The most important words in the topic (i.e. those with the highest probabilities) are the largest in the word cloud, and 20 words are shown in total.
- The ten questions with the highest proportion of that topic in their topic distribution. This helps ground the topic in real questions and clarify any ambiguities caused by the

word cloud.

If the respondent lacks enough knowledge on either of the two topics (or on both), they are asked to choose an option indicating that. They are asked (but not required) to provide a free-text rationale for each of their responses, and they also have to explain concisely what they understand as the concept of each topic, which can help filter out the responses where the respondent has clearly misunderstood the topic. The responses for the difficulty question are as follows:

- Topic A much more difficult than topic B
- Topic A slightly more difficult than topic B
- Equal or incomparable
- Topic B slightly more difficult than topic A
- Topic B much more difficult than topic A
- I lack knowledge on at least one of these topics.

The responses to the survey have been collected through crowdsourcing, using the platform Prolific. This platform allows for filtering users out based on the industry they work in, and only users with “software” indicated as their industry were allowed to answer the survey. This step is used, combined with the screening questions and a manual review of the free-text explanations given by the respondents, for guaranteeing that the respondents really *are* software developers and that they have understood the task correctly. In addition, the respondents were asked to enter their field of expertise and programming experience. In order for a user’s responses to be eligible for manual screening, they had to answer at least one of the 3 screening questions correctly – i.e. indicate the correct topic as the more difficult one – while not answering any of them in the wrong direction (i.e. indicate the wrong topic as the more difficult one). This allowed for some “equal or incomparable” responses and also “I don’t know” responses in the screening questions.

Once the responses are collected, the relationship between the developer perceptions and the estimated difficulty can be investigated, and the level of agreement existing among developers themselves can be quantified. Since perceived difficulty is a subjective concept, a certain degree of disagreement is expected, but too much disagreement runs the risk of making the results random. Therefore, investigating the level of agreement is as important as investigating the correspondence between estimated and perceived difficulty.

Topic model, Score/Topic count	60	80	100	120	140	160	180	200
CTM, Log likelihood	-6.6205	-6.5806	-6.5555	-6.5269	-6.5090	-6.4940	-6.4796	-6.4593
LDA, Log likelihood	-6.6948	-6.6552	-6.6213	-6.5957	-6.5766	-6.5489	-6.5382	-6.5218
CTM, Exclusivity	9.9239	9.9459	9.9535	9.9642	9.9682	9.9710	9.9780	9.9787
LDA, Exclusivity	9.8425	9.8654	9.8806	9.8937	9.9015	9.9089	9.9148	9.9197

Table 3.1 – The log likelihood and exclusivity scores for the CTM and LDA models trained on the sampled question set. In case of both scores, higher is better.

3.3.3 Research questions

To summarize the goals of this study, the research questions can be presented as follows:

RQ1: What are the main factors affecting question response times on Stack Overflow, and how much of the differences in response times can be explained by the topics of a question? Are there significant differences between topics in terms of their effects on response times?

RQ2: What is the perception that developers have of the relative difficulty of a pair of topics, and how correlated is this perceived difficulty with the estimated relative difficulties?

RQ3: How much do developers agree on the perceived relative difficulties of topics?

3.4 Results

3.4.1 Topics and question response times

Fitting the topic model

The first step of the data-driven pipeline in this study is fitting a topic model to extract topics from Stack Overflow questions. For this task, a subsample is created of the Stack Overflow questions posted in the 2014-2018 period. The questions from before 2014 are eliminated in order to ensure that more *current* Stack Overflow topics are identified, and a subsampling scheme is used to favor less popular and more obscure tags, so that the extracted topics are not overwhelmingly biased towards popular ones. In order to create this subsample, all the questions from the aforementioned period with a negative Stack Overflow score are first eliminated. Then, a weighting scheme similar to inverse document frequency (IDF) is used, where the weight of the question q, denoted by w_q, is equal to:

$$w_q = \sum_{t \in tags(q)} log \frac{N_q}{N_{q,t}}$$

where N_q is the total number of questions (from the selected period) and $N_{q,t}$ is the number

Feature	Overall effect (sign)
Responder: accepted ans. min. time	+
Responder: accepted ans. median time	+
Responder: # of all ans.	+
Answer: body length	+
Q: posted on weekend	+
Inquirer; # of all answers to questions	–
Question: code to text ratio	–
Answer: # of URLs	–
Inquirer: first ans. median time	–
Responder: Sum of answer downvotes	–

Table 3.2

of questions that have the tag t. This scheme assigns low weights to questions on very popular topics, thus allowing questions on less popular topics to make up a larger proportion of the sample of question. Then, 500,000 questions are sampled using the weights assigned to them. This number of questions is more than enough for the topic model, but again allows less popular topics to be more prominent than they would be if larger numbers of questions were sampled. After sampling the questions, their text undergoes several standard preprocessing steps (e.g. code and link removal, stemming, and stopword removal) and is prepared for the extraction of topics.

To extract topics, the R package STM (Roberts et al., 2019) was used in order to fit CTM and LDA models with various numbers of topics (from 60 to 200) to the sampled questions, and the log likelihood scores (on a holdout set of 10,000 questions) and exclusivity scores can be seen in Table 3.1. Based on these scores (and the declining rate of score improvement as the number of topics goes up) and a manual investigation of the topics, a CTM with 180 topics was chosen as the best model . Then, in order to clean up the topics and make sure that they are reasonably human-readable and have captured meaningful topics, a manual investigation of the topics was conducted. Two groups of features were removed: a few that had captured very general topics unassociated with any particular skill (e.g. topics whose most important words are “problem”, “help”, “google”, etc.), and a few that were very noisy and not human-readable (which also had very low proportions across all questions). The result was a topic model incorporating 166 topics, which was renormalized to make the sum of every document’s topic proportions equal to 1.

Top 5 words in topic	Topic summary	Effect (sign)
test,unit,mock,spec,suit	Unit testing	+
sample,layer,network,busy,rate	Neural networks	+
login,authent,token,secure,author	Authentication	+
build,release,built,gradle,jenkin	Build tools	+
driver,mongodb,spark,embed,hadoop	Hadoop	+
array,numpy,want,one,2d	Numpy	−−
layout,width,height,center,vertical	Basic CSS	−−
string,separator,split,quote,character	Basic strings	−−
column,row,dataframe,want,sum	Pandas	−−
list,want,tuple,one,example	Basic Python iterables	−−

Table 3.3 – The top 5 (largest positive effect) and bottom 5 (largest negative effect) topics in the response time classifier. All of these topics have a statistically significant effect on response time class.

Fitting the classifier

In order to fit the classifier to a recent set of questions, all the questions created in the year 2017 with an accepted answer and with a non-negative Stack Overflow score were taken (for a total of 910,312 questions). Then, two response time classes were created by taking the 20% of the questions with the highest response times and also the 20% of the questions with the lowest response times. The former are "late-answer" questions (for which the *smallest* response time is 16 hours), whereas the latter are "early-answer" questions (for which the *largest* response time is 10 minutes). The elimination of the middle 60% is done because response times have a considerable degree of inherent randomness, and only keeping the extremes can help reduce the impact of this randomness by making sure that there is a drastic difference between response times in the two classes. The logistic regression model is then trained to predict the response time class of each question[1]. This model achieves an F1-score of 0.76, which means that its performance is reasonable and it is possible to gain insights into the effects of the features on the output by investigating their weights in the model. At a significance level of 0.00025 (which is the original level of 0.05 with a Bonferroni correction to account for the number of statistical tests, which is around 200), 42 features in total achieve significance (i.e. their effect values are significantly different from 0), with 16 of them being topics. The ten features with the greatest positive and negative contributions to response times can be seen in Table 3.2. As the results clearly show, the most important features in the model are the confounding features pertaining to both the posts (question and answer) and to the users (inquirer and responder). At the same time, about 40% of the significant

[1] Needless to say, the model only predicts the response time class for the questions that were in the top or bottom 20% of response times, since the middle 60% have already been discarded.

features are topics, showing that the topics of a question do have effects on its response time that cannot be explained by other factors. The five topics with the largest positive effects and the five topics with the largest negative effects are shown in Table 3.3. The topic-agnostic popularity feature does not achieve significance, which shows that the actual content of the question matters more than how popular its topics have been independently of the content. More importantly, the pairwise Wald's tests for *differences* between the effects of topics (at a significance level of 0.000001, again Bonferroni-corrected to account for the large number of statistical test) finds significant differences in about 1200 of the pairs. This means that the study can proceed with the second step, feeding a sample of 35 significantly different topic pairs and 35 non-significant topic pairs to the developer survey.

3.4.2 Developer survey results

A total of about 90 responses were received from developers. Out of these, almost half (44) had to be filtered out due to mistakes in the screening questions, undutiful answers to the survey (i.e. all answers being the same or all the answers being "I don't know"), giving the wrong description of the concept of the topic, or not giving an "I don't know" response when the description indicated lack of knowledge. This resulted in 46 times 7 (equalling 322) real question responses (i.e. excluding the screening questions) to 70 questions, averaging 4-5 responses per question, which is a reasonable number even though larger numbers would have been more desirable.

Analysis of multiple-choice responses

To analyze the multiple-choice responses in the survey, the respondent's answer to each question is put into one of the following three response categories (unless it is an "I don't know", in which case it is discarded):

- **Positive**: The respondent's answer **agrees** with the estimated relative difficulty. For example, topic A is estimated to be more difficult than topic B, and the respondent's answer is that A is more difficult.
- **Neutral**: The respondent's answer is that they are equal or incomparable in difficulty.
- **Negative**: The respondent's answer **disagrees** with the estimated relative difficulty. For example, topic A is estimated to be more difficult than topic B, but the respondent's answer is that B is more difficult.

The questions themselves were grouped into two categories: those where the topic pair had a significant difference (at a significance level of 0.0001) in estimated difficulty, and those where the topic pair did not have a significant difference. The binary indicator variable for these categories shall be called “significance category”, with 1 indicating a significant difference and 0 indicating a lack thereof. In order to see whether or not significance category had a significant effect on the response category, a Chi-squared test (with $\alpha = 0.0001$) was conducted, which found a significant effect. However, although the significance category did have a significant effect on the response category, the developers did not agree with the estimated difficulty when it came to the topic pairs that had a significant difference in estimated difficulty. To clarify, the effect of the significant category on the response category simply means that the developers agree *more* with the estimated difficulty difference when that difference is significant, but it does not say anything whether or not the developers agree with the difficulty estimation per se. Converting the multiple-choice responses into integers from -2 to +2 (for the five possible responses, with the same logic as negative/neutral/positive from before), the mean response to the significantly different pairs is **-0.12**, which means that developers generally tend to *disagree* with the estimated difficulty! As a point of comparison, this mean value is **-0.29** for the non-significant pairs, while in the ideal case, this value would have been around zero for the non-significant pairs while being a positive value for the significantly different pairs. To summarize, the estimated difficulty seems to correspond weakly to the opposite of perceived difficulty.

Furthermore, and perhaps more importantly, the developers were found to have wildly different perceptions of the relative difficulties of the presented topics. In order to measure the degree of agreement among developers, their responses were made ternary (i.e. the “much more difficult” and “more difficult” options were taken to be the same, resulting in 3 difficulty comparison options instead of 5) after removing the “I don’t know” responses, and Krippendorf’s Alpha was computed on the resulting responses, yielding a value of 0.09. This value means that the developers’ responses are *almost random*. In the literature, in labelling tasks, a Krippendorf’s Alpha below 0.66 is often taken as a sign that the labelling task is problematic and the respondents have been unable to agree enough on the labels, while a value of 0 means that the responses are entirely random (Hughes, 2021). Therefore, a value of 0.09 is not very far from entirely random responses, and means that there is an undesirably large degree of ambiguity and subjectivity in the task, potentially both in the topics themselves and in the question of perceived difficulty. In order to understand why this is the case, it is necessary to investigate the free-text responses given by the respondent as their rationales.

Analysis of textual responses

The textual responses of the respondents have been manually categorised into seven categories. Then, a “contradiction” between two rationales (which can be between two different arguments or the *same* argument) was defined as the two being used as responses to the same question,

Rationale	Usage count	Contradiction count (self-contradictions counted twice)
Technical knowledge or solution complexity	92	54
Availability of resources (documentation and online Q&A)	13	4
Mathematical knowledge prerequisite	12	2
Debugging and errors	12	8
Superset or prerequisite relationship	11	6
Incomparable or equal in difficulty	78	-
Personal experience	48	-

Table 3.4

but for giving contradictory answers (i.e. one is used for saying that A is more difficult than B, while the other is used for the opposite). Self-contradictions (i.e. the two rationales being the same) count twice. These rationales are shown, along with both their usage counts and their total contradiction counts in Table 3.4 (the respondents who gave no textual responses at all are excluded from this table). To count contradictions, the "incomparable" responses and the responses with "personal experience" rationales have been removed, thus only keeping the responses that *can be* contradictory (i.e. the ones that are not neutral) and where the respondent tried to give a relatively objective rationale. This is why those two types of rationale have no contradiction counts. There may remain a certain amount of implicit subjectivity in the rationales, which cannot be qualified using the data available in the survey responses.

The "technical knowledge or solution complexity" rationale is the most common, and as such, it is also unsurprisingly the one most commonly used in contradiction with others and itself, being used contradictorily a total of 16 times against others and 19 times against itself (counting self-contradictions only once). This means that it is used in a non-contradictory fashion 38 times (a total of 92, plus 16 times contradictorily against others and 19 times against itself, which means a total of 16 + 2*19 contradictory uses), which is only 41% of the time. It is also the only rationale that is used in contradiction with itself – other rationales have no self-contradictions in the collected survey responses. Also notably, the "debugging and errors" rationale appears 8 times in contradiction with the "technical knowledge or solution complexity" rationale, meaning that its non-contradictory use happens only 33% of the time. These numbers, in addition to the number of explicitly subjective responses, point towards additional subjectivity in the rationales, potentially stemming from personal experience, expertise levels, and also ambiguities in the topics. In particular, a manual investigation of the descriptions given by users about the topics in each question reveals ambiguities in the topics that led to users having different perceptions of each topic, especially in how general or specific they consider the subject to be, pointing to the importance of further clarification of what each topic really stands for.

3.5 Discussion

3.5.1 Implications of results

The first basic finding of this study is that although the topics of Stack Overflow questions play a significant role in whether or not they will go unanswered for a long time, there is no clear relationship between the perceived difficulty of Stack Overflow topics and the response times of their questions. This lack of a clear relationship is due to a combination of the following:

- The ambiguity of the topics extracted from the questions, leading developers to have different perceptions of what the topics mean, particularly when it comes to how general or how specific the topics are.
- The subjective and experience-based nature of perceived difficulty, which is based on how the person learned the topic in the first place, and what they have struggled with the most. Previous work on perceived difficulty has focused on a much smaller scale and on well-defined, short tasks, rather than entire topics, and this scale partially explains the large amount of observed disagreement in this study.
- The ambiguity of the survey question itself, which may have led some developers to state the difficulty they experienced when *learning* the topic while others stated the difficulty they had *using* the topic's relevant skill. This is especially problematic since these two processes, i.e. learning vis a vis using, are tangled when it comes to self-learning (which is what Stack Overflow is mainly used for), further complicating the question.
- The relatively low number of responses, which made it difficult to use the auxiliary information such as years of experience and specific areas of expertise.

Therefore, undeniably, the results of this study have shown that the relationship between perceived difficulty and response times is tenuous at best. This means that the different effects that topics have on response times on Stack Overflow should be attributed to factors other than difficulty. These factors could include the following:

- How time-consuming (rather than difficult) it is to answer the average question in the topic. In particular, the *types* of questions asked in different topics may differ, e.g. one topic may have more debugging questions, while another may have more "how to" questions.
- How large and experienced the sub-community is for that topic. This is a very important factor, because the fewer the "Stack Overflow experts" for a particular topic, the more likely it is that the harder questions on that topic would go unanswered for a while.

The second and more important finding of this study is the almost complete disagreement of developers about the perceived difficulties of topics, which complicates things even further. If developers had greater agreement on the responses, and the only problem was that no significant relationship had been found between response times and perceived difficulties, a redesign of the approach and the survey could have been the solution. However, this finding has an more fundamental implication for the remainder of this dissertation: if perceived topical difficulty is so subjective and contradictory, then it is not a suitable direction for research on training needs. In other words, despite the ease of going from estimated topical difficulty to insights for training providers (as discussed early in this chapter), the estimation of perceived topical difficulty on a large scale may be borderline infeasible in the software programming domain as the opinions of developers are simply too contradictory. Since perceived difficulty is the easiest type of difficulty to measure (as it can simply be crowdsourced, compared to performance-based difficulty which requires hard-to-obtain performance data, as discussed in the second chapter), this essentially rules out the use of difficulty in this dissertation, and thus, the rest of the chapters will move away from this approach.

3.5.2 Refinements and future work

Despite the failure of this methodology and the move away from it for the rest of this dissertation, it is important to discuss the ways in which this study could be made more rigorous and potentially more effective, even if a move away from Stack Overflow response times would be necessary.

First of all, the difficulty rationales given in the survey can be used to create a refined survey where each difficulty rationale is a *separate question*. For example, one question would ask the respondent whether one topic is more difficult than the other because it has less training material available, while another would ask whether they have a difference in difficulty because one requires more prior knowledge to learn. This could be paired with an alternative framing of the main question, such as “how many hours of work would it take for a novice to learn this topic”. This can help the respondents think about the topics in clearer terms and give less contradictory responses, although the reliability of the individual responses would remain an issue, and large numbers of responses would be required to fully take advantage of these refined questions.

Secondly, to clarify the topics and reduce their ambiguity, it may be useful to further refine the topic model and merge some topics in order to eliminate the issues that occur because of the varying specificity of the topics. In fact, in hindsight, the use of a large number of topics may have contributed to greater ambiguity in this study, and a balance needs to be struck between coverage (i.e. covering as many topics as possible) and ambiguity. Closely related to this improvement is the better representation of topics. For Stack Overflow (or any other Q&A

platform), this could be done by sampling the questions representing the topic in a way that showcases both the depth and the breath of the topic. Thirdly, instead of asking the user about their expertise at the end, the survey needs to ask the user regarding their *certainty* about each of their responses, so that a confidence score can be assigned to each respondent's answer to a particular question. This can allow the researcher to assign more importance to the responses of those respondents with more expertise, although the confidence scoring scheme needs to also account for the fact that different respondents have different levels of self-confidence.

Finally, as the related work section showed, the concept of perceived difficulty has proven useful when applied to more specific topics on smaller scales (i.e. one language, rather than every topic in a professional domain). Therefore, one way to potentially improve the pipeline and get better results is to focus on one specific group of topics, e.g. web development. This would reduce the variety of topics and would also make the Stack Overflow expert communities smaller and more overlapping, thus potentially contributing to less ambiguous and subjective responses to the developer survey.

3.6 Conclusions

A data-driven methodology was proposed combined with a developer survey to investigate how the latency of question responses on Stack Overflow is related to the difficulties of their topics for developers. The results indicate that there is no clear link between perceived difficulty and question response times, and that generally, profound disagreement exists among developers about what makes a topic difficult and about the relative difficulties of those topics. The failure of this methodologies has guided the work in the present dissertation away from difficulty-based approaches and towards approaches analyzing the evolution of the skills landscapes of professional domains.

4 Study 2: Dynamics of Online Hiring and Learning Platforms

4.1 Introduction

In the previous chapter, the ideas behind the first study and its failure were discussed in detail. The most important takeaways were the fact that Stack Overflow on its own was not a sufficient data source for understanding training needs, and that too much reliance on expert or practitioner opinions could lead to too much subjectivity and disagreements. As discussed in the previous chapter, these factors, combined with a realization that the approach was too indirect to spend more time on, contributed to the abandonment of the previous, difficulty-based approach.

The innovation diffusion model, however, remains a very helpful conceptual model to base our work on, as we are still interested in the early life of a new skill, and not its life after it has achieved mass adoption. In addition, Stack Overflow remains a potent data source for understanding the trends of various software programming skills, as it provides a view of the educational demand for those skills. This view however, is only one possible view into the landscape of education and recruitment, and it is only with a more thorough study of this landscape that the full picture can be obtained. The “early identification of innovations” goal that forms the backdrop of every study in this dissertation would be better served by such a full picture, as the innovations involved (which could be new skills or technologies) could manifest on a variety of platforms and data sources.

In this second study, the aim is to investigate and understand the early life of new skills and technologies in the software programming profession by looking at four online platforms:

1. Stack Overflow, already discussed in detail;
2. Google Trends, a tool that provides the normalized Google search volume of a given query over time;

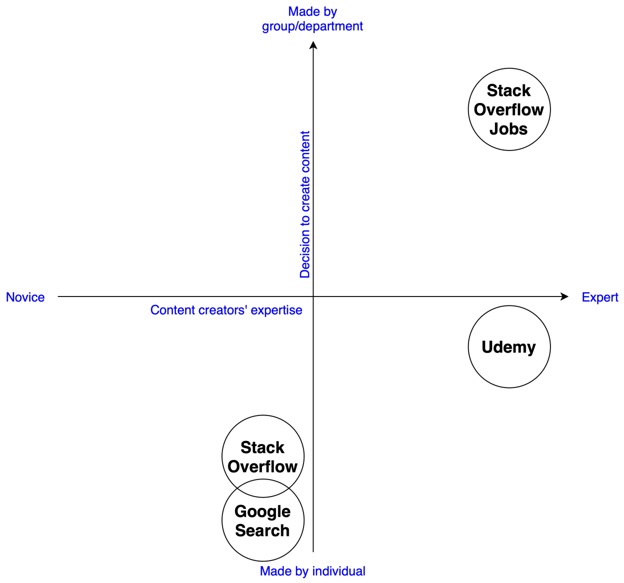


Figure 4.1 – The four online platforms in our study, positioned along two axes. The horizontal describes how much subject matter expertise the *average* content creator has, while the vertical describes how, on average, the content creation decision is made: individually, or by groups/departments.

3. Udemy, a Massive Open Online Courses (MOOCs) platform where anyone can create and share a free or paid MOOC, and where MOOCs are organized around the practical skills to be acquired by students;

4. Stack Overflow Jobs, a job ad platform created on the same domain as Stack Overflow for software development-related jobs.

The new skills and technologies in this study are identified through (and represented by) Stack Overflow "tags": user-created words or phrases that are used to describe the topics of Stack Overflow questions. Tens of new tags are created on Stack Overflow every week[1], allowing them to serve as a crowd-sourced representation of new topics in the software domain. To understand the early life of new topics and to measure the agility of each platform, this study analyzes and compares the times at which each new tag shows up on each of the four platforms for the first time. The hypothesis is that two factors contribute to a platform being more agile, i.e. manifesting new topics earlier: lower expertise and effort being needed for the creation of content[2] (or, in other words, when novices and experts alike can create

[1] Also, over 60,000 tags already exist on Stack Overflow, with their popularity (i.e. number of questions having the tag) more or less following a power law.

[2] It is necessary to clarify here that "creating content" for a platform like Google Trends simply means searching for a query and thus generating data. For Udemy and Stack Overflow Jobs, it is posting a course/lecture or a job ad,

content), and decisions to create content being made more individually, rather than by groups or departments. These four platforms can be placed along the two axes of “content creator expertise” and “individuality of content creation decision making”, as shown in Fig. 4.1, and this study investigates the degree to which the hypothesis that “more novice-driven and individual-driven platforms are more agile” holds. This is the core of the **descriptive** part of this study. The next hypothesis is that *if* some platforms *systematically* manifest new topics earlier than others, then perhaps the appearance of those new topics on the latter platforms could be predicted using signals from the former. This forms the **predictive** part of this study.

The results of the study show that in the majority of cases, the first hypothesis - that platforms where content creation is less expert-driven and more individual are more agile - holds true. However, as we will see, the aforementioned “majority” is not overwhelming, and some topics appear first on the more expert-driven and/or less individual-driven platforms. It was also found that the software programming profession is very agile as a whole, with the median delay between the first appearance of new topics on Stack Overflow and their first appearance on Udemy or Stack Overflow Jobs (i.e. the more expert-driven platforms) being around 3 or 4 months, respectively. Some variance is observed in this delay and in the proportion of tags where the first hypothesis holds true, based on the subject matter and granularity of the tag (e.g. whether it is about web development or cloud computing, or whether it is a language or a framework, etc.)[3]. Regarding the second, prediction-related hypothesis, it was found that early appearance signals based only on user activity on the more agile platforms are insufficient for predicting the appearance of new topics on the less agile platforms. The results of this study quantify the agility of various online platforms in the software programming profession and confirm Stack Overflow’s position as the most agile, demonstrate the variation of this agility when it comes to different groups of topics, and also demonstrate the rising agility of Udemy, which has become an immensely popular MOOC platform. The proposed methodology is generalizable, and given the right data sources, it can be used to analyze other professional domains, allowing training program creators across many domains to better understand the speed at which their target domain evolves and to focus their attention on the important areas. It also provides a base for expansion, allowing it to serve as a basis for deeper dives into a particular professional domain, especially software programming. Therefore, this study’s focus on helping training providers in adapting their curricula to the changing skill landscape can help expand the scope of educational research to include its economic context, which is a documented shortcoming of previous educational research Greer and Thompson (2016); Carnegie and Crane (2019).

The majority of this chapter’s content is taken from the paper “Keeping Up with the Trends:

respectively; while for Stack Overflow, the definition of content creation here is limited to posting questions, for reasons that will be explained in the Methodology section.

[3] As you may have noticed, Google Trends is missing from this discussion: that is because its extreme data sparsity in the very specific topics under study rendered it effectively unusable for this study.

Analyzing the Dynamics of Online Learning and Hiring Platforms in the Software Programming Domain", which we published in the International Journal of AI in Education. The introduction and conclusion have been modified to integrate the paper's contents with the contents of the thesis, and the related work section has been shortened to avoid too much repetition of what was already explored in Chapter 2.

4.2 Related Work

As stated before, much of the relevant literature has already been discussed in Chapter 2's "Related Work" section, so this section will only provide a refresher on the big data sources involved in this study, as they are especially relevant here and warrant repetition.

4.2.1 The role of Big Data in the new economy

As discussed before, the belief that big data will serve as a transformative force in the economy is prevalent among academics (Horton and Tambe, 2015; Einav and Levin, 2014). New data sources such as online educational and hiring platforms provide new opportunities for empirical research on labor markets, and on skills in particular (Horton and Tambe, 2015). The important advantage of these online data sources over administratively curated data sources, such as national income and employment data, is that the latter, despite their quality and breadth, are very hard to collect and are therefore collected infrequently (Horton and Tambe, 2015). Big data sources, however, are continually updated and can provide extremely granular data on each user. Job ad platforms such as LinkedIn, Q&A platforms such as the Stack Exchange family, and MOOC platforms such as Coursera and Udemy are all part of this emerging set of data sources (Horton and Tambe, 2015). The educational platforms mentioned, in particular, are geared towards lifelong learning, and they especially enable informal e-learning Latchem (2016), although technologies such as Q&A forums can also be used to support more formal learning Hammond (2019). These new, large, broad, and continuously updated data sources have enabled research that is more fine grained than what was possible before. Studies on firms' human capital investments in IT in general Tambe and Hitt (2012) and big data in particular Tambe (2014), or studies of labor flow among organizations Tambe and Hitt (2013) are all examples of this.

Of course, with these opportunities come many challenges. These challenges include 1) problems of data acquisition from firms who would guard them as their property, 2) issues stemming from a biased selection of users, either through a researcher's sampling or due to the nature of the data itself, and 3) challenges in processing the data, especially given the mismatches that exist between different data sources (Horton and Tambe, 2015; Einav and Levin, 2014). In addition, such analyses may require metrics and methods different from what is usually used (Einav and Levin, 2014). It is worth stressing that when it comes to making economic predictions (which is one of the aims of this study), Einav and Levin (Einav and Levin, 2014) are less enthusiastic, based on the fact that predictive models are often not "structural", meaning that they do not learn the underlying processes — and these processes react to prediction-based (or rather any) policy changes. This means that policy changes could result in behavioral changes in the system, which would mean that the formerly accurate predictive model would no longer be well-suited to the system.

4.2.2 Big Data sources: online hiring and learning platforms

In the literature review in Chapter 2, the wide variety of digital platforms for education were discussed. Given the focus of this study on online hiring platforms, Q&A platforms, and MOOC websites, it is useful to discuss the literature surrounding these data sources again, particularly to position this study within the existing literature. Readers of Chapter 2 may find significant portions of this section familiar, and are advised to only read the last paragraph of each of the following subsections, which are concerned with positioning.

Online hiring platforms and job ad collections

Analyses of massive numbers of job ads are the most prevalent type of analysis when it comes to understanding labor market trends, as they manifest the skills demanded by employers. Many such analyses are conducted by corporations that collect or host this type of data (effectively on a yearly basis) (BGT, 2019; LinkedIn, 2019). Some analyses of job ads only analyze what is *currently* in demand, and do not focus on trends and changes in the skill landscape (Hiranrat and Harncharnchai, 2018; Papoutsoglou et al., 2017). The analyses that focus on job ad trends (Strack et al., 2020; LinkedIn, 2019) are often thorough in analyzing various types of skills. In particular, the Boston Consulting Group, in partnership with Burning Glass Technologies Strack et al. (2020) distinguish between emerging skills, i.e. skills that used to have a small market share but are growing very rapidly, versus fast-growing skills whose share of the market used to be considerable already.

All of these studies only use job ads, and therefore only analyze the big picture through the lens of hiring platforms, ignoring the potential of educational platforms to manifest skill trends earlier. This is of particular concern for training program creators, since having a head start would allow them to prepare their material in advance and to have them already prepared once the skill has become more trending.

Online question answering platforms

Recent years have witnessed a dramatic rise in the popularity of Q&A websites such as the Stack Exchange family, and in particular Stack Overflow, which is a platform for software developers. These platforms have moved away from simply providing good answers for the question askers, and towards becoming repositories of community-curated knowledge. (Anderson et al., 2012). A considerable body of research exists on Stack Overflow, tackling subjects such as the content of questions and their trends (Barua et al., 2014; Allamanis and Sutton, 2013), the interactions of Stack Overflow with other platforms such as GitHub (Vasilescu et al., 2013), and many others. Stack Overflow's popularity and dynamics, such as the high level of moderation present on the website (Correa and Sureka, 2014; Ponzanelli et al., 2014) and the quality control measures, make it an invaluable data source when analyzing the software industry.

Despite the significant number of studies on Stack Overflow, few of them present methods for analyzing the topics discussed there and their trends (Barua et al., 2014; Yang et al., 2016; Ishola and McCalla, 2016). Most of these (Barua et al., 2014; Yang et al., 2016) do not make a particular attempt to study *new* topics on Stack Overflow; instead, their methodology involves training time-independent topic models (like Latent Dirichlet Allocation Blei et al. (2003)) and tracking the popularity of each of the identified topics over time. Ishola and McCalla Ishola and McCalla (2016), on the other hand, present a methodology to track the knowledge needs of a learner on Stack Overflow, using Stack Overflow tags as their "evolving knowledge ontology". However, their methodology focuses on the evolution of the learners themselves, and does not focus on the evolution and the trends of the skills that are to be learned.

Massively Open Online Courses

Online learning has witnessed a quick rise to prominence in recent years, owing to their usefulness as an instrument of lifelong learning, and much scholarly work has been conducted on these platforms (Ebben and Murphy, 2014; Conache et al., 2016; Zhu et al., 2018; Bozkurt et al., 2016). Some MOOC platforms, such as Coursera, tend to mimic actual universities: they have academic schedules, with exercises, quizzes and exams, and the possibility of earning certificates for single courses or for entire programs. On the other end of the spectrum are MOOC platforms that offer individual skill-based courses, such as Udemy, with self-paced and skill-centric MOOCs. Udemy is of particular note because its business model is one where any person can sign up to become a content creator and create (free or paid) MOOCs for others to use (Conache et al., 2016); this is in contrast to platforms such as Coursera, Udacity and edX, where the content providers are universities, organizations or corporations(Conache et al., 2016).

Much like the work on online hiring platforms and job ad collections, most of the skill trend analysis here comes from the MOOC platforms themselves, such as Coursera Coursera (2019) and Udemy Udemy (2020). Indeed, most existing MOOC research focuses on the students' experience, their motivation, their retention, and on the design and assessment aspects of the MOOC (Zhu et al., 2018; Bozkurt et al., 2016). Studies such as (Coursera, 2019) and (Udemy, 2020) use student enrollment data (e.g. the time series for the number of people enrolled in each course) to understand trends, and focus on the most trending skills, the most popular skills, the differences between different geographical regions, with more in-depth analyses of certain technologies. Again, similar to studies on hiring platforms, these studies make a deep dive into the topic of skill and technology trends, but only utilize one source of data, meaning that combining them with other types of data sources (like data from Q&A platforms and hiring platforms) would let them paint a more comprehensive picture.

4.2.3 Positioning the study

The review above sketched the landscape of research efforts that are relevant for this study, as summarized in Table 4.1. This study aims to exploit the opportunities created by the availability of relevant data sets in order to complete this landscape. Many scholars focus on formal learning environments, including formal MOOCs, but this work explores less formal contexts such as a Q&A system or Udemy (less formal MOOCs). Also, many scholars investigate job markets through various recruitment platforms or Stack Overflow. The originality of this study lies in the exploration of the relationship between multiple data sources that are actually connected by a common societal cycle: training in specific skills, recruiting staff with specific skills and practicing these skills (in this case, asking for help while working). Here, the relationship between platforms is studied by focusing on time, i.e. the question of when specific skills / topics appear in one platform as compared to other platforms. Identifying this temporal relationship paves the road towards predicting training needs. Such a prediction, even if marginally successful, could be a huge advancement in understanding emerging skills and making education and training institutions more agile.

4.3 Research methodology

4.3.1 Platforms and datasets

As mentioned before, the data analyzed here come from four different sources.

1. The software development Q&A platform "Stack Overflow" is the most important data source, for two reasons: first, it is massively popular and usually ranks very high in web search queries; second, it allows users to create *tags* that indicate the topics of a question, letting them put up to 5 tags on each question they post, showing the various aspects

Subject	Relevant state of the art (in descending order of relevance)	Existing gaps
Job ad platforms	Skill trends Emerging skills Skill demand ...	No dynamics with edu platforms
MOOC platforms	Course/skill demand Course design Student interactions with MOOC ...	Small focus on skills no time dynamics
Q&A platforms (Stack Overflow)	General skill trends Question content Interactions of Stack Overflow with GitHub User interactions ...	No focus on new skills Need more inter-platform dynamics

Table 4.1 – A summary of the relevant state of the art and existing gaps in the literature on the platforms that have been deemed useful for this study. The reasons why this study only looks at these three types of platforms are detailed in Section 4.2.2.

of the question and connecting it with all the other questions they share a tag with. For example, the very popular question “What does the ’yield’ keyword do?” (which is about the ’yield’ keyword in the Python programming language) has the tags “python”, “iterator”, “generator”, “yield” and “coroutine”, each describing the question’s topic from a certain aspect and at a certain granularity, linking the question to all others that share a tag with it. In addition to asking and answering question, users can vote for, or comment on a question/answer posted by someone else, providing an interesting dynamic of user interactions. All historical data are available as a download from The Internet Archive[4]. In this study, the primary objects of interest are the questions, their posting dates and their tags. The Stack Overflow question has a central role, since each comment, answer, or vote can be traced back to exactly **one** question, making the question the centerpiece of each chain of interactions.

2. “Google Trends” provides Google search volumes for any search term during different periods of time. The search volumes provided by Google Trends are normalized to 100, meaning that the largest value retrieved for any Google Trends query (for any period and any term) is 100 (and as such, raw search volumes cannot be retrieved from Google Trends). It is a potentially important data source in and of itself, since searching on Google could very well be the most popular way of looking for an answer to a programming question. In addition, it reveals two indicators of interest that cannot be seen on

[4] https://archive.org/download/stackexchange

Stack Overflow: on one hand, duplicate questions are not allowed on Stack Overflow (so each question can only be *asked once*), and on the other hand, a person could refer to the same question multiple times. These two indicators of interest are invisible on Stack Overflow, since Stack Overflow only keeps the latest number of views for a post, and there is no "view count history" available for its questions. Since Google Search is a popular way of finding the relevant Stack Overflow question, the two aforementioned indicators of interest could be indirectly observed on Google Trends. This dataset is obtained through an unofficial Python library that acts as an ad-hoc API[5] to Google Trends. The main objects of interest here are the search volumes over time for various topics. Since access to raw Google search data is unavailable, and since Google's method for computing the query counts is not public, the amount of interpretation that can be done using this dataset is limited. As we will see later, Google Trends does not do well when it comes to very granular topics, and fails to prove useful for the topics studied in the present work.

3. "Udemy" is a MOOC platform where anyone can create and publish a MOOC. Udemy is a highly popular platform with over ten thousand software development related MOOCs, from short and specific courses (e.g. a crash course on setting up an Amazon Web Services server) to broad and in-depth courses that cover entire jobs (e.g. web development bootcamps). Each course consists of lectures, quizzes, and exercises, although the overwhelming majority (about 87 percent) are lectures. There are over 10,000 software development courses on Udemy, and the Udemy course creator community is decentralized and made up of many people. It is reasonable to expect that this decentralization could let them recognize trends and react to them more quickly than a more traditional MOOC platform where decision-making is centralized. Through the developer API[6], openly available data on Udemy courses may be obtained. The data of interest here are the creation and publication dates of courses, their titles, and their syllabi, which consists of lecture titles and the creation date of each lecture.

4. Stack Overflow Jobs is a job ads platform integrated with Stack Overflow, where many employers advertise available positions for developers. The employers provide a description for the job, tag the job ad with the previously mentioned Stack Overflow tags, and provide other details such as the job's salary and benefits. Many companies are present on Stack Overflow Jobs, with over 18,000 unique companies having posted ads as of December 2019. Although not available directly as a download on The Internet Archive, this dataset can be obtained through the RSS feeds of its pages scraped daily (and often multiple times a day, necessitating duplicate removal) by the Wayback Machine[7],

[5] https://github.com/GeneralMills/pytrends
[6] https://www.udemy.com/developers/affiliate/
[7] https://web.archive.org/web/*/stackoverflow.com/jobs

Dataset	Relevant data	Earliest data
Stack Overflow	Tags and timestamps for each question	Sep. 2008
Stack Overflow Jobs	Publisher, tags, and description for each job ad	Oct. 2015
Udemy	Course and lecture titles and timestamps for each course	May 2010
Google Trends	Normalized search volume time series for each term	Jan. 2004

Table 4.2 – A summary of the relevant data in the data sources used in this study, along with the earliest date for which data can be obtained from each data source.

which is the web scraping bot of The Internet Archive. From this dataset, the title and description of each ad, its posting date, and its tags are the data that matter in this study.

These four data sources, summarized in Table 4.2, provide different perspectives into the topics that exist in the software programming domain (and this study deals with the new topics among them). Stack Overflow and Udemy are two (very different) types of educational platforms, Stack Overflow Jobs is a hiring platform, and Google Search essentially serves as an educational platform as it facilitates access to educational material. As we already saw in 4.1, these four platforms are placed along the two axes of "content creator expertise" and "decision to create content", with the former going from "novice" to "expert" and the latter going from "made by individual" to "made by group or department". The four platforms are placed on this 2-dimensional plane based on their *average* content creator:

- Questions on Stack Overflow may be created by a novice looking for more basic information, or by an expert asking an advanced question. However, the quick response times observed on Stack Overflow Bhat et al. (2014) make it likely that the majority of question-posters are asking rather "easily-answerable" questions. Therefore, Stack Overflow is placed leaning towards the novice side (but not much), and pretty much entirely on the "decision made by individual" side, since ultimately, it is one person deciding whether they have a question or not.

- Google Trends stands at approximately the same place as Stack Overflow. This position is not fully accurate, but for the purposes of this study, this much accuracy would suffice. This is because Google Trends and Stack Overflow are effectively never compared in this study (and what is more, Google Trends did not prove to be as useful as we had hoped).

- Udemy courses have to be made by experts, but the course or lecture creation decisions are not constrained to being made only by individuals: they could also be made by small teams (e.g. if the course is a joint venture by several instructors).

- As Stack Overflow Jobs is a hiring platform, the people creating the ads have to be subject matter experts (or have consulted with experts), and they are part of HR departments

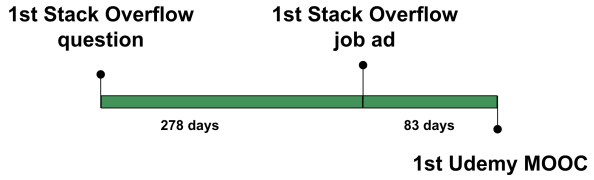


Figure 4.2 – The delays between appearance on Stack Overflow, Stack Overflow Jobs, and Udemy for the tag vuejs2, which is about Vue.js version 2, a Javascript framework.

that represent entire organizations, meaning that the decisions cannot have been made individually. Therefore, this platform lies at the extreme upper right part of the plane.

4.3.2 Aims and Methods

So far in this chapter, topics (which are proxies for skills, which would be training needs) have been discussed, and the way the four platforms provide different views on them has been laid out. This part provides a discussion of this study's general methodology for identifying and studying those topics, and the way in which the aims of the study shape the methodology will be explored.

Aims

The goal of this study is twofold. The first aim is to understand the early life (and in particular, the first appearance) of new topics on the four platforms discussed previously, looking at how agile each platform is, how the platforms evolve over time, and how differently they behave with respect to different topics. The second aim is to see if the appearance of these topics on the expert-driven and more group/department-driven platforms could be predicted using signals from the more novice-driven and individual-driven platforms. From now on, the first aim shall be called the **descriptive** aim and the second shall be called the **predictive** aim. Additionally, the two clusters of platforms will be referred to as expert/group-driven and novice/individual-driven, respectively, although individual discussions of the platforms are also included. In order to work towards these two aims, let us define and then investigate the following:

- Appearance ordering: The order in which a new topic appears on the four platforms. For the prediction aim, this is very important because for all the topics that appear first on the more expert/group-driven platforms, the prediction task using the novice/individual-driven platforms is rendered pointless. Fig. 4.2 shows the tag vuejs2 (for the Javascript framework Vue.js, version 2), having appeared first on Stack Overflow, then on Stack

Question	Tags
Why is processing a sorted array faster than processing an unsorted array?	java; c++; performance; optimization; branch-prediction
How do I undo the most recent local commits in Git?	git; version-control; git-commit; undo; pre-commit
Implementing a Neural Network in Haskell	algorithm; haskell; neural-network; backpropagation
Debugging in Safari's Web Inspector, when using a module loader like SystemJS	javascript; safari; ecmascript-6; systemjs; web-inspector

Table 4.3 – Examples of Stack Overflow questions and their tags for four questions. The first two are two of the most popular questions with accepted answers, while the last two are two of the most popular questions without accepted answers.

Overflow Jobs, and finally on Udemy.

- Prediction window: The time between the first appearance of a new topic on any novice/individual-driven platform, and its first appearance on any expert/group-driven platform. This tells us how much time a training provider, wishing to preempt the expert/group-driven platforms, would have for creating a course or lecture in the ideal case – the case where it would be possible to predict any topic's eventual importance correctly, right when it appeared on a novice/individual-driven platform. Multiple criteria are used in this study for the prediction window in order to also gauge the strength of the predictive signals. Fig. 4.2 shows the prediction windows for vuejs2 for Stack Overflow Jobs (278 days) and for Udemy (361 days).

- Topic "themes" and "types": A topic's **theme** indicates what field of software development it falls into (e.g. web, cloud computing, mobile development, etc.), while the **type** indicates whether it is a concept or a product, and in the latter case, what kind of product it is. Examples of topic types include frameworks, full-fledged solutions, libraries within a language, concepts, etc. The present investigation of the early life of new topics on the four studied platforms is not complete without an investigation of how different *kinds* of topics differ in their behavior on these platforms.

Topics and Tags

So far, the goals of this study with respect to the analysis of "new" topics have been established. It is now time to discuss what will be used to represent topics in the software programming domain: Stack Overflow tags. As mentioned before, Stack Overflow tags are user-created words or phrases that indicate the topics of a question, and a user posting a question can tag it with up to 5 of these tags. A few more examples of Stack Overflow questions and their tags can be seen in Table 4.3. In addition, Stack Overflow moderators can mark tags as synonyms, hold polls for deleting them, or change the tags of a question to make them more appropriate if the original tags are inadequate. This makes Stack Overflow tags a set of precise and specific community-curated topics, and when a question has a tag, it means that the question is relevant to the skill that the tag corresponds to (e.g. "python" corresponds to the Python programming language, while "iterator" corresponds generally to using iterators in loops). As a result, for the purposes of this study, Stack Overflow tags are a good proxy for skills. It is important to acknowledge that this approach mostly excludes higher order skills such as 'agile development practices' or 'redesigning a client-server architecture for minimizing bandwidth requirements', as these usually do not appear as tags on Stack Overflow.

In an ideal world, a Stack Overflow user would create a new tag *only* if no existing tag described their specific question accurately enough, therefore making the tags almost perfect representatives of new and emerging topics, with perfect precision (i.e. every new tag describes a new topic) but potentially imperfect recall (i.e. not every new topic is immediately the subject of a question) that would improve over time. In this world, however, there are two potential issues:

1. A user could create a duplicate, unimportant, or otherwise redundant tag.
2. A tag could be created so late that its topic would not be considered "new" anymore. This includes a topic never showing up as a tag on Stack Overflow.

These two issues impact the "descriptive" and "predictive" goals of this study differently. The first issue is significant given that tens of new tags are created every week, and the only way to resolve it automatically at scale is to wait for community signals of tag quality and importance, such as whether or not the tag becomes popular later on. This solution is an impediment to the predictive aim, since it is desirable to perform predictions as early as possible. However, it does not impact the descriptive aim as much, as the descriptive aim in fact *requires* studying a longer period of time, so that a wider range of tag behaviors can be observed. The second issue is only a problem if the tag appears late on *all* of the platforms under study (since it is possible - and as we will see, quite common - for tags to appear first and early on Udemy or Stack Overflow Jobs) *or* does not appear on Stack Overflow at all. In the former case, it would mean that the topic is not new anymore, while in the latter case, the topic's existence would never be detected by our methodology. This issue is therefore very difficult to address, but given Stack Overflow's popularity, it is reasonable to assume that a new topic never appearing on Stack Overflow is very unlikely, given that this study looks at a long period of time. Again, this second issue is more of a problem for the predictive aim, because as mentioned before, for a tag that does not appear on novice/individual-driven platforms before the expert/group-driven platforms, the prediction task is meaningless. As a result, these two issues are not addressed for the predictive aim of this study. For the descriptive aim, however, a partial solution will be introduced later in this section in the form of tag **popularity**, helping make the results of the study more robust.

Before proceeding further, it is necessary to discuss the choice of Stack Overflow tags as the representatives of topics in this study, rather than topic modelling approaches such as Latent Dirichlet Allocation (LDA) Blei et al. (2003) (even though the latter may be better at capturing higher-level skills). There are three reasons for this choice:

1. Tags are user-created and also edited by moderators. Therefore, a great amount of crowdsourced manual work has gone into them, and the use of tags makes use of already-done work. Performing topic modelling from scratch would require finding the optimal number of topics, and then interpreting the topics and figuring out what each one really means. As seen in Study 1, such an approach can be problematic and can require significant manual effort Chang et al. (2009).

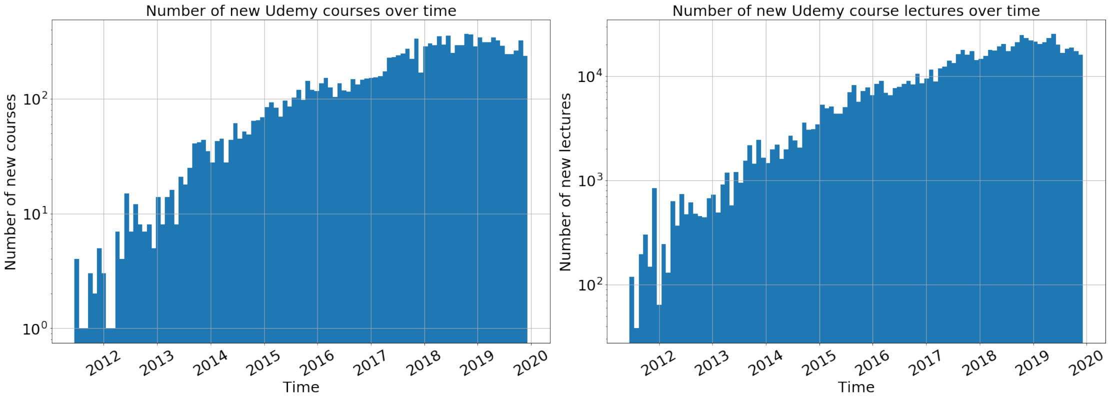
2. Stack Overflow tags can be very specific, e.g. the tag "laravel-5.8" refers to version 5.8 of the web framework Laravel. Topic models usually have trouble representing such fine details, and are mostly useful for gaining higher-level and more general information on a corpus Wallach et al. (2009).

3. Tags are exact matches, meaning that a tag either *is* found in a course, job ad, or question, or *is not*. However, topic models are probabilistic Blei et al. (2003), meaning that each document is expressed as a distribution over topics, which are themselves distributions over words. Therefore, determining the threshold at which a document is considered to "contain" a topic becomes a problem in itself.

4. As the previous chapter showed, topic modelling lacks transparency, whereas tags are much more transparent and require less interpretation.

Connecting the datasets

In order to track the same topics across the four data sources, the following steps are performed for each tag:



(a)

(b)

Figure 4.3 – Number of new courses (a) and lectures (b) published on Udemy over time (until Dec. 2019), divided into 100 time bins. Each bar is the number of new courses/lectures published during that time bin's duration. The vertical axis is logarithmic in scale.

- Its popularity on Stack Overflow is observed, in the form of question counts and vote count time series, in particular looking at the earliest question that has the tag.
- Its Google Trends search volumes are retrieved.
- Its occurrences in the lectures/course titles of Udemy MOOCs are found.
- Its occurrences in the descriptions and tags of Stack Overflow Jobs ads are found.

In order to find occurrences of each Stack Overflow tag in the titles of Udemy course syllabi and in job ad descriptions, a direct matching is performed to find the occurrences of each tag in the text. An alternative would have been to train a machine learning model to classify a piece of text as containing or not containing a tag. However, given the sheer number of tags and the differences of text styles between job ads, courses, and questions, this could lead to a considerable error potential, thus making the direct matching approach less error-prone. The details of the matching method can be found in Appendix B. After performing the matching, 'generic' tags such as "introduction" are removed from the list of tags. These tags are neither technologies nor concepts, and therefore are not relevant to this study. Generally, Stack Overflow has a policy of removing (or in their own terminology, burninating) these generic tags, but many have yet to be removed. Fortunately, it was found that most of these tags appear on Udemy and Stack Overflow Jobs during the early phase of the platforms' existence, and thus can be removed them by eliminating all the tags that have appeared on these platforms during their early life. It is now time for a discussion of warm-up periods.

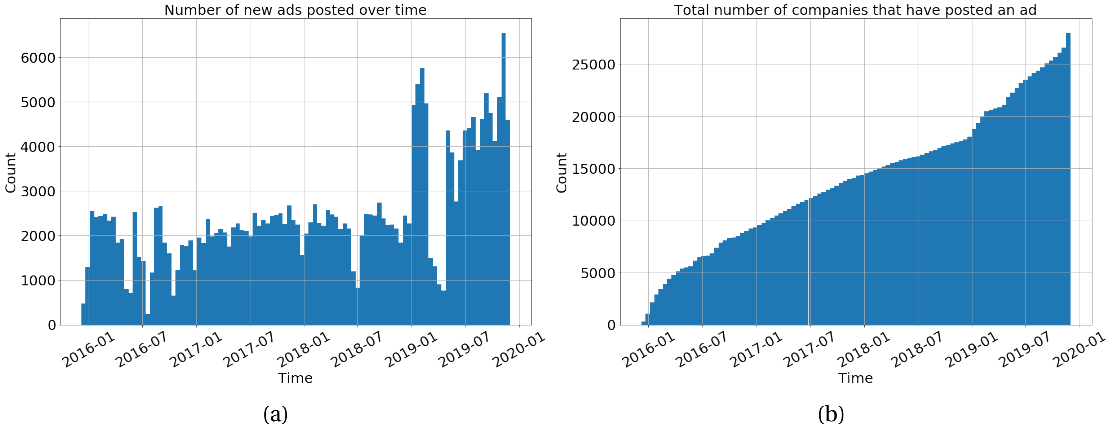


Figure 4.4 – Number of new ads published (a) and the total number of companies that have posted an ad so far (b) on Stack Overflow Jobs over time (until Dec. 2019), divided into 100 bins. Each bar is the number of new job ads published during that time bin's duration. The total number of unique companies at by the beginning of January 2019 is 18095.

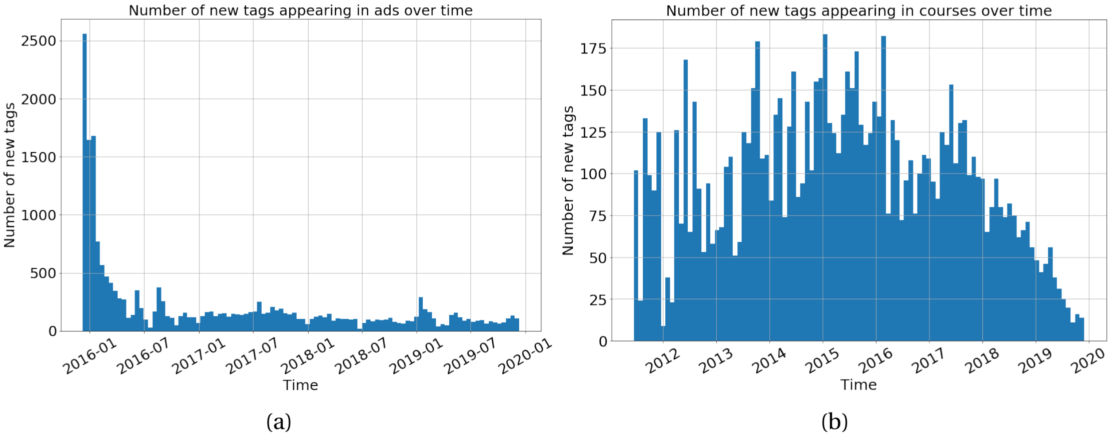


Figure 4.5 – Number of new tags appearing on (a) Stack Overflow Jobs and (b) Udemy over time, divided into 100 bins. Note the considerably different vertical scales and also horizontal scales.

Starting dates and platform warm-up

An important question in this work is the question of the periods under study. Stack Overflow started its operation in 2008, Udemy in 2010 and Stack Overflow Jobs in 2015. Hence, all those tags that have appeared on Stack Overflow before 2010 have trivially appeared there prior to their appearance on Stack Overflow Jobs and Udemy. Unaccounted for, this could lead to a fallacious confirmation of the hypothesis that the tags appear on Stack Overflow before Udemy or Stack Overflow Jobs. In addition, each platform has had a **warm-up period**, during which it was only beginning to gather momentum. While a platform is still in its warm-up, many of the tags that appear on it are not *new* topics; they are rather topics that have existed for a long time and are already well-known and important. Therefore, they do not indicate a training need arising from the emergence of a technology or skill, but are rather a sign that people are starting to recognize and use *the platform itself.*

Fig. 4.3 shows histograms of new course and new course lecture counts on Udemy from the beginning until December 2019. Similarly, Fig. 4.4 shows the histogram of the number new job ads published in Stack Overflow Jobs since its creation in late 2015 until December 2019, divided into 100 bins. These two histograms serve the purpose of demonstrating the platforms' respective growths in popularity and content, in particular showing us when the growth has stabilized. The end of the warm-up periods of these two platforms is identified by looking at the stabilization of the number of new tags appearing on each platform. Figure 4.5 shows a histogram of new tag counts over time for both platforms. Based on these figures, the end of the warm-up period for Udemy has been fixed at July 2013 (which right after the jump seen in Fig. 4.3a) and for Stack Overflow Jobs at October 2016 (which is again after a jump, seen in Fig. 4.4a). In order to have fair comparisons between the different datasets, when connecting several datasets together, a **starting date** is chosen that is after the end of every dataset's warm-up period, and all the tags that have appeared on *any* of the platforms before that date are removed. This has the added advantage of removing the generic tags that mostly appear during a platform's early life.

In addition, as shown in Fig. 4.4, Stack Overflow Jobs has suddenly experienced a much higher level of popularity in 2019, which could indicate that a new phase in the platform's existence has begun, where many new companies are starting to use the platform and many more ads are being posted. In order to avoid treating the two clearly different phases the same way, and (quite importantly) because of how short the second phase is (only 11 months before the study was conducted), this study has been limited to tags created *on every single platform* before January 2019. The reason for enforcing this end date on all the datasets is to be fair towards all the datasets, since the focus is on the time at which a tag manifests on *each dataset* for the first time. Therefore, only data generated up until the end of 2018 has been studied.

"Popular" tags

As mentioned when introducing tags as the proxy for topics, there are two situations where the argument that "new tags represent new topics" falls short. Those issues are harder to address for the predictive aim, as this aim requires quick action and does leave much time to wait for community action on the tags. For the descriptive aim, however, there is more time, and it is thus possible to address these issues using signals from the community. Here, the concept of "popular" tags comes in. When it comes to reasonably popular tags that have been used on many Stack Overflow questions, the Stack Overflow community has essentially confirmed their quality and importance. Therefore, studying these tags could resolve the issue of unimportant or duplicate tags, and help verify whether the aforementioned problem could invalidate this study's hypotheses. To this end, "popular" tags will be defined, and the descriptive measures of the study will be reported separately both for "all" tags and for "popular" tags only.

To define "popular" tags, two measures are computed regarding the early life of each tag:

1. Total number of questions on the tag in its first 365 days of existence (or all of its existence if it is younger than 365 days), divided by 365 if the tag is older than 1 year, and divided by its age in days otherwise.
2. Total number of votes (upvote and downvote) for the questions on the tag in its first 365 days of existence (or all of its existence if younger than 365 days), again divided by 365 if it is older than a year, and divided by its age in days otherwise.

These two measures indicate how popular the tag was/is in its first year of existence, while not being unfairly biased against tags that are less than one year old at the end of the studied period (i.e. the end of 2018). "Popular" tags are defined as those that have at least 10 questions in total, and for which at least one of the two measures above is in the top 50% among all Stack Overflow tags created between July 2013 and January 2019[8]. The medians for our two popularity measures are 32 questions per year and 32 votes per year, and the aforementioned criteria result in a total of 5575 popular tags. Of course, this definition is not set in stone, because there is no real "ground truth" on popularity, but it is reasonable to believe that the combination of the two measures mentioned above, along with the conduct of two analyses (popular tags and all tags) will result in a reasonably accurate picture of the different behaviors of tags.

4.3.3 Research questions

Summing up the discussions of this study's methodology and aims, and based on the definitions given earlier, the research questions can be given as the following:

- **RQ1:** Are some appearance orderings systematically more common than others? Do tags tend to appear first on novice/individual-driven datasets before appearing on expert/group-driven datasets, and if so, to what degree?

[8] These two dates are the warm-up date of Udemy and the end date of this study. The tags created on Stack Overflow between these two dates are all the tags that *could* be matched to at least one of the other datasets, and thus could be part of the tags in this study.

- **RQ2:** How long is the usual prediction window between the appearance of tags on novice/individual-driven and expert/group-driven platforms?
- **RQ3:** How do appearance orderings and prediction windows vary among different categories of tags, both in terms of their themes and in terms of their types?

4.4 Results

4.4.1 Appearance Ordering

The first research question is concerned with the order in which new tags *appear* on the four platforms, so let us go over what this *appearance* means for each platform. For Stack Overflow, Udemy and Stack Overflow Jobs, its definition is simple: the creation of the first question with the tag, the first course or lecture mentioning that tag in its title, and the first ad that has that tag in either its description or its list of tags, respectively. For Google Trends, the first appearance is when the value of the normalized search volume goes above 0 for the first time[9].

Most tags trivially appear on Stack Overflow, but their existence on the all other three platforms is not a given. Therefore, the unit of study is a **dataset groups**, wherein the only tags considered are those appearing on all the datasets in the group (e.g. one data set are the tags that appear on Stack Overflow and on Udemy). These different groupings are interesting to study because they reveal different dynamics between the datasets. In addition, given the definition of a dataset group given above, larger groups of datasets would most likely have fewer tags. This makes investigating less restrictive groupings (i.e. with fewer datasets in the group) quite important, because as we will see, the number of tags can get quite small.

Table 4.4 should be read as follows: each row is a **dataset group**, with a **starting date** as defined before, and a binary indicator of whether the row considers only popular tags, or all tags. This means that the tags considered in that row are those that appeared on *every* dataset in the group and did so *after* the starting date. If the row's name says "popular", then out of the aforementioned tags, only those that also meet the popularity criteria are considered; the name will say "all" otherwise. Each percentage column (whose name starts with %) shows the percentage of tags that appeared on the column's dataset before appearing on any of the others. Therefore, for example, the cell that is the intersection of "SO, Udemy, July 2013, all"[10] and "% SO first" gives us the percentage of tags in this group that appeared on Stack Overflow before appearing on Udemy. Since the hypothesis under investigation for RQ1 is that tags have a tendency to appear on more novice/individual-driven platforms before more

[9] In order to retrieve the specific tag on Google Trends, the tag is treated as an n-gram: dashes are replaced with whitespaces and quotation marks are placed around the query, in order to make sure that no unrelated search queries are included through individual words in the query.

[10] In our tables, SO is Stack Overflow and GT is Google Trends.

Row #	Dataset Group	# tags	% SO first	% Udemy first	% SO Jobs first	% GT first
1	SO, Udemy, July 2013, all	1368	78.07***	21.93	-	-
2	**SO, Udemy, July 2013, popular**	**731**	**84.95***	**15.05**	**-**	**-**
3	SO, Udemy, Oct. 2016, all	215	71.16***	28.84	-	-
4	**SO, Udemy, Oct. 2016, popular**	**117**	**77.78***	**22.22**	**-**	**-**
5	SO, SO Jobs, Oct. 2016, all	648	64.66***	-	35.34	-
6	**SO, SO Jobs, Oct. 2016, popular**	**167**	**85.63***	**-**	**14.37**	**-**
7	SO, Udemy, SO Jobs, GT, Oct. 2016, all	88	56.81	23.86	17.05	2.28
8	**SO, Udemy, SO Jobs, GT, Oct. 2016, popular**	**57**	**63.16***	**21.05**	**12.28**	**3.51**
9	Udemy, SO Jobs, Oct. 2016, all	518	-	55.02	44.98	-
10	**Udemy, SO Jobs, Oct. 2016, popular**	**206**	**-**	**58.74**	**41.26**	**-**

Table 4.4 – Number of matched tags for each “dataset and starting date” group, along with the percentage of X-first tags in that group, where X is each of the four datasets. SO stands for Stack Overflow and GT stands for Google Trends. The rows where only popular tags are considered are in **bold**. The null hypothesis of the statistical test performed (Pearson’s Chi-Squared) on every row is that the combined percentage for the novice/individual-driven datasets is equal to the combined percentage for the expert/group-driven datasets (and equal to 50%), implying random ordering between the two. The signs *, **, and *** (shown in the “% SO first” column) correspond to the three significance levels 0.05, 0.01 and 0.001, respectively. The statistical test is inapplicable for the last two rows.

expert/group-driven platforms, a statistical test has been performed for each row. This test is a Pearson's Chi-Squared test, with the null hypothesis that the percentage of tags appearing first on the novice/individual-driven platforms is *equal* to the percentage of tags appearing first on the expert/group-driven platforms (which would mean that the distribution is 50-50).

Several important observations can be made from Table 4.4 regarding **RQ1**:

1. Generally, the skills landscape of the software programming domain evolves very quickly. This can be seen by looking at the number of matched tags in of the rows of Table 4.4. For example, there are 648 tags matched between Stack Overflow and Stack Overflow Jobs starting from October 2016, 167 of which are popular. Since the data are from October 2016 to the end of December 2019, this means about 199 new tags per year, with about 51 popular new tags per year. Looking at the match between SO Jobs and Udemy, and the match between Stack Overflow and Udemy gives different numbers (63 and 36 popular new tags per year, respectively), but in all these cases, there is a considerable number of new topics every year, even if some of them are relatively small topics (e.g. versions of an existing software or programming language, or a specific topic within some language). This is especially important from the educational perspective: each year, there are many trends for educators have to keep up with.
2. **The null hypothesis of the statistical test in Table 4.4 is rejected in all but one case**, but the rejection gets very weak (or fails outright) in rows with more datasets matched together (and hence fewer matched tags). It is strongly rejected when matching Stack Overflow to Udemy or Stack Overflow Jobs separately (rows 1 through 6), but it becomes weakly rejected (for popular tags) or unrejected (for all matched tags) when all the datasets are matched together (rows 7 and 8). This could be a result of the greatly reduced numbers, but it is undeniable that the percentage of SO-first tags does drop considerably when all the datasets are matched together. In addition, **across the board, the "popular" tags (even-numbered rows) have greater SO-first (and generally, novice/individual-driven-first) percentages, compared to "all" tags (odd-numbered rows)**. This is particularly striking when matching SO and SO Jobs, where close to three quarters of all the matched tags are "unpopular", and the SO-first percentage is much higher among the "popular" tags. Since, as discussed before, the "popular" tags have exhibited greater post-hoc importance (in the form of user interest on Stack Overflow) and thus are more relevant to the descriptive aim, this lends greater support to the hypothesis behind RQ1.
3. **Even though the null hypothesis is rejected in most cases, the percentages are far from 100%**. As mentioned before, the prediction task is rendered meaningless for any tag that appears on Udemy/Stack Overflow Jobs before appearing on Stack Overflow/-Google Trends. This, combined with our results, means that even optimistically speaking,

the prediction task is meaningless for around 30% of the tags. Viewed another way, this would mean that in around 30% of the cases, the “experts” already knew the importance of a tag before this study’s method could get a chance to notify them of it. Note again, that the SO first percentages, as mentioned before, are lower for “all” tags compared to “popular” tags.

4. **In all the dataset groups where both Udemy and SO Jobs are present, Udemy comes first more frequently than SO Jobs**. In the direct comparison between rows 9 and 10, with a starting date of October 2016, the first Udemy course/lecture comes before the first SO Jobs ad in more than 50% of the cases, although not much more than 50%. A Pearson’s Chi-Squared test, with a null hypothesis of the ordering between Udemy and SO Jobs being random (i.e. 50-50), rejects the null hypothesis with $p < 0.01$ for both “all” and “popular” tags. Despite the relatively small difference in percentages, even parity in the ordering between the two would have been an interesting result, since large educational institutions that rely on committees for their decision-making (e.g. universities) are generally not very quick to react to market trends. This, therefore, serves as evidence that Udemy is remarkably agile in creating courses, which is to be expected given its model of “anyone can create a course”.

5. **When it comes to reducing the number of tags, limiting the set of tags to popular tags has a more pronounced effect on the results for Stack Overflow Jobs than for Udemy (see rows 5 and 6)**. Also, the difference in the proportion of SO-first tags between rows 5 and 6 is staggering: limiting the tags to the popular ones increases the SO-first percentage by over 20% (compared with only about 7% for Udemy in rows 3 and 4)! This shows that many of the tags that are used on Stack Overflow Jobs are not so relevant or important to the Stack Overflow developer community. According to the difference observed between rows 3 and 4 versus rows 5 and 6 in the table, the appearance of a tag on Stack Overflow Jobs is a weaker sign of the tag’s actual importance as a topic on Stack Overflow, compared to its appearance on Udemy, although more of the popular tags appear on Stack Overflow Jobs than on Udemy (which could be attributed to the fact that Stack Overflow Jobs shares tags with Stack Overflow).

In all of the dataset groups investigated above, Google Trends was very rarely the first place where a new tag appeared, being surpassed by every other dataset by a large margin. Only one example has been included in Table 4.4, with other examples skipped for brevity. An investigation of the reasons behind this revealed that many of the tags under investigation had a volume of *zero* on Google Trends, due to their extreme specificity. Part of this is because of the use of quotation marks around n-gram tags for looking them up on Google Trends, which was done because of two reasons: 1) to avoid ambiguities arising from polysemous words, and 2) because the exact way Google Trends handles queries with multiple words is not clear (e.g.

	Stack Overflow-first	Udemy-first	Stack Overflow Jobs-first
Popular	**angular5,** **pytorch**	**google-cloud-firestore,** **coreml**	**azure-kubernetes,** **go-ethereum**
Non-popular	codemod, smack-stack	docker-app, zig	mix-and-match, neptune

Table 4.5 – Examples of popular and non-popular tags that appeared on Stack Overflow, Udemy, or Stack Overflow Jobs first. Popular tags are in bold as established in the previous table.

we searched for the tag "vuejs2" as "Vuejs 2", in quotation marks). This ends up eliminating many relevant queries along with the irrelevant ones, resulting in very low search volumes. Therefore, for a study of specific technologies and concepts with very specific names and various versions, Google Trends has limited usefulness. In the results in the future sections, Google Trends has been excluded for this very reason.

To summarize the results of this section, the hypothesis in RQ1 is found to be true, especially when "popular" tags are concerned, and there is a systematic tendency for tags to appear first on novice/individual-driven platforms and then on expert/group-driven platforms. The percentages observed also indicate that this is not the case for all tags, and that there is a limit on the potential effectiveness of our prediction task. Table 4.5 shows, for each of the three categories of Stack Overflow-first, Udemy-first, and Stack Overflow Jobs-first, two examples of popular tags and two examples of non-popular tags that fall into that category.

4.4.2 Delays and Prediction Windows

To answer the second research question, summary statistics (1st quartile, median, 3rd quartile) are calculated for the prediction windows of tags in various dataset-starting date groups, shown in Table 4.6. In addition to the delay between the 1st Stack Overflow question and respectively the 1st Udemy course/lecture and the 1st Stack Overflow Jobs ad, the time between the 1st and 5th Stack Overflow questions of each tag is also studied. The appearance of the 5th question is an event that implies rising popularity, but is still very commonplace and happens for many eventually unimportant tags. Out of the 6676 tags created on Stack Overflow after October 2016 (and before January 2019), 3489 have at least 5 questions. This is a much greater number than the number of tags matched between Udemy and Stack Overflow that also have 5 questions (215 tags), or Stack Overflow Jobs and Stack Overflow (648 tags), for the same starting date.

First of all, the results show that with the October 2016 starting date, the lead Stack Overflow has on Udemy and Stack Overflow Jobs is not large: the median delay from the first Stack Overflow question to the first course/lecture is around 3 months for "all" tags and around 4 to 5 months for "popular" tags. This median delay is greater for the job ads, with the "popular" tags

Dataset Group	**1st question to 5th question**		**1st question to 1st lecture**		**1st question to 1st ad**		**1st lecture to 1st ad**	
	# tags	quartiles	# tags	quartiles	# tags	quartiles	# tags	quartiles
SO, Udemy, July 2013, all	1261	q1: 54 q2: 139 q3: 313	1368	q1: 48 q2: 402 q3: 813	-	-	-	-
SO, Udemy, July 2013, popular	**731**	**q1: 36 q2: 84 q3: 224**	**731**	**q1: 133 q2: 446 q3: 793**	**-**	**-**	**-**	**-**
SO, Udemy, Oct. 2016, all	184	q1: 23 q2: 66 q3: 153	215	q1: -42 q2: 92 q3: 281	-	-	-	-
SO, Udemy, Oct. 2016, popular	**117**	**q1: 18 q2: 41 q3: 88**	**117**	**q1: 21 q2: 141 q3: 303**	**-**	**-**	**-**	**-**
SO, SO Jobs, Oct. 2016, all	464	q1: 40 q2: 98 q3: 205	-	-	648	q1: -114 q2: 119 q3: 327	-	
SO, SO Jobs, Oct. 2016, popular	**167**	**q1: 20 q2: 46 q3: 89**	**-**	**-**	**167**	**q1: 91 q2: 244 q3: 406**	**-**	
SO, Udemy, SO Jobs, Oct. 2016, all	80	q1: 18 q2: 43 q3: 114	88	q1: -76 q2: 86 q3: 293	88	q1: -14 q2: 172 q3: 386	88	q1: -131 q2: 93 q3: 246
SO, Udemy, SO Jobs, Oct. 2016, popular	**57**	**q1: 15 q2: 39 q3: 75**	**57**	**q1: -24 q2: 125 q3: 300**	**57**	**q1: 13 q2: 237 q3: 489**	**57**	**q1: -98 q2: 132 q3: 250**
Udemy, SO Jobs, Oct. 2016, all	-	-	-	-	-	-	518	q1: -172 q2: 33 q3: 208
Udemy, SO Jobs, Oct. 2016, popular	**-**	**-**	**-**	**-**	**-**	**-**	**206**	**q1: -147 q2: 44 q3: 213**

Table 4.6 – Number of matched tags, and the three quartiles of event delays (**in days**) for different dataset-starting date groups. Rows for popular tags only are in **bold**. For each column, positive quartile values mean that the first event in the column's name came first, while negative values mean that the second event happened first. For example, among all of the 215 tags matched between Stack Overflow and Udemy with a starting date of October 2016 (the third row in the table), the 1st quartile, median and 3rd quartile of the delay between the 1st question and the 1st Udemy course/lecture are -42, 92, and 281 days respectively.

having medians as high as around 8 months. However, in both cases, the delay is a fraction of a year, and in particular, the delay of 3 to 5 months that we see for Udemy is quite short. This is because of the fact that a course or lecture also takes time to create; a median of 3 months - from the *first* Stack Overflow question to the first course/lecture - means that even if the importance of the tag is predicted correctly on day *one*, the content creator receiving this information will still only have 3 months to act on it – that is, if they want to be the creator of the *first* course/lecture.

Secondly, an interesting pattern can be observed: the values for "all" tags are almost always (and in case of the median, always) lower than for "popular" tags. This means that when considering the "popular" tags, Stack Overflow always has a larger lead on Udemy and Stack Overflow Jobs. Our hypothesis for explaining this observation is that in general, the studied tags are user-created, with no prior expert vetting; therefore, for those that do not show *evidence* of importance, their getting matched to MOOCs or job ads may not mean much, and may occur more erratically. It is the "popular" tags for which there is greater evidence of importance, and therefore, they provide more reliable information when it comes to the goal of **describing** the behavior of the platforms in this study.

The third interesting observation is the difference between the statistics for the groupings between Stack Overflow and Udemy, for the two starting dates of July 2013 and October 2016. As can be seen in the first four rows of Table 4.6, the quartiles are much higher for the July 2013 starting date, with the October 2016 medians being around four-fold smaller than the July 2013 medians. This is a very interesting finding, and begs the following question: has Udemy become more agile in responding to new topics, or has Stack Overflow lost its agility?

The agility of Stack Overflow and Udemy

To answer the question of the agility of these two platforms, it is necessary to look at the delays between the 1st and 5th questions on Stack Overflow, for the two starting dates of July 2013 and October 2016 in Table 4.6. It is clear that the delay between the 1st and 5th questions has almost been reduced to half, and thus the new tags seem to get their 5th question much more quickly. Although this does not directly mean that the first appearance of the tag on Stack Overflow is also happening more quickly, it is nevertheless an argument in favor of Stack Overflow having become *more* agile, not less. Therefore, there is no reason to believe that Stack Overflow has experienced a reduction in agility. This lends support to the hypothesis of Udemy's increased agility. In order to properly ascertain this increased agility, two sets of tags are created:

1. Tags created on both Stack Overflow and Udemy between July 2013 and October 2015.

Tag set (Udemy + Stack Overflow)	**1st question to 5th question**		**1st question to 1st lecture**	
	# tags	quartiles	# tags	quartiles
July 2013 to October 2015, all	230	q1: 54 q2: 171 q3: 413	250	q1: -41 q2: 191 q3: 383
July 2013 to October 2015, popular	**127**	**q1: 35 q2: 79 q3: 239**	**127**	**q1: 58 q2: 259 q3: 437**
October 2016 to January 2019, all	184	q1: 23 q2: 66 q3: 153	215	q1: -42 q2: 92 q3: 281
October 2016 to January 2019, popular	**117**	**q1: 18 q2: 41 q3: 88**	**117**	**q1: 21 q2: 141 q3: 303**

Table 4.7 – Number of matched tags, and the three quartiles of event delays (**in days**) for the two sets of tags matched between Stack Overflow and Udemy, used to analyze the agility of the two platforms. As always, the rows that only consider popular tags are in **bold**. Notice that from the earlier set to the later set of tags, the median 1st question to the first lecture time has dropped from 191 days to 92 days for all tags, and from 259 days to 141 days for popular tags, representing an almost 2 fold reduction. This is while the 1st question to 5th question time has also experienced a very considerable drop.

2. Tags created on both Stack Overflow and Udemy between October 2016 and January 2019.

These two sets of tags cover the older and newer periods of Udemy's life, and have the same length of 27 months; this helps avoid potential biases arising from one set covering a longer period of time, and thus having a greater likelihood of having larger delays[11]. Statistics on the delays can be seen in Table 4.7. In order to see whether the distribution of the delay between the 1st question and 1st course/lecture is significantly different for sets 1 and 2, the Mann-Whitney U test is used. The null hypothesis of the test is that the delay of a tag randomly sampled from one set is equally likely to be greater or smaller than the delay of a tag randomly sampled from the other set. The test's null hypothesis is rejected both for the two sets of tags, and for their two "popular" subsets (in both cases with $p < 0.01$), implying that the distributions of delay are indeed different. This combination of evidence means that Udemy *has* indeed become significantly more agile over the years, and the time it takes for the Udemy creators' community to react to new topics in the software industry has been shortened considerably, almost by a factor of 2. It is unclear whether this increased agility is

[11] This is because the delay distribution is clearly biased towards positive delay values.

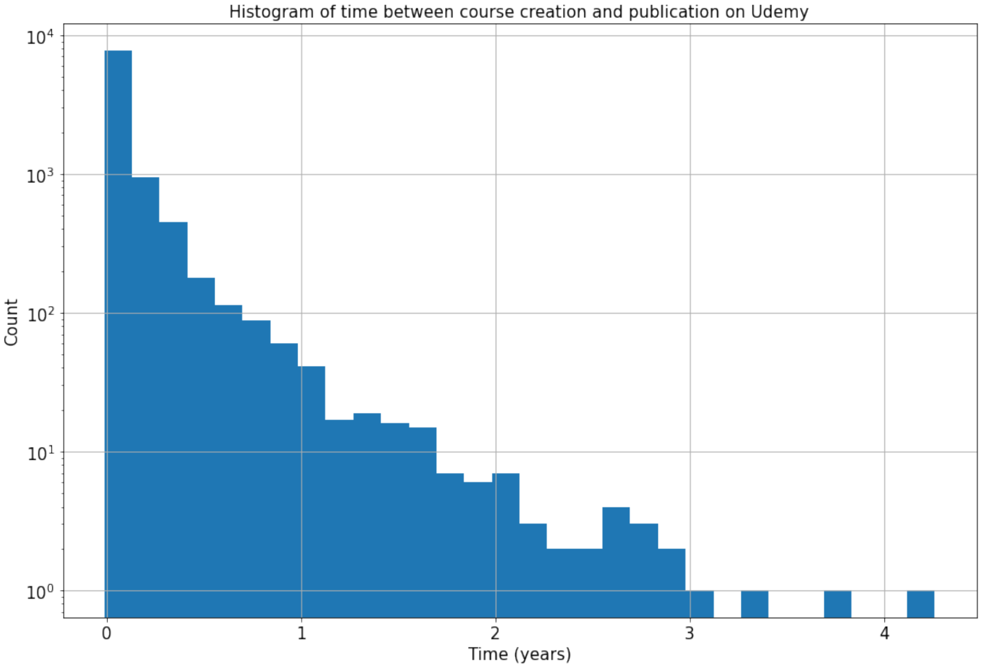


Figure 4.6 – Histogram of time (in years) between course creation and publication on Udemy. Course creation effectively amounts to "reserving a spot" for the course, while publication is akin to making it available to users.

only the result of an ever-expanding creator and student community, or if Udemy's policies and algorithms have also independently contributed to it, but the (rather sparse) existing literature and history available on Udemy do not give us much other than basic statistics on Udemy Conache et al. (2016) or its content recommendation systems Wai (2016), and credit is likely due to the expanded creator and student communities for Udemy's increased agility.

On the topic of Udemy's agility, it is useful to also look at statistics on how quickly Udemy courses are made. A useful statistic here is the difference between the time when the course was **created** and the time when it was **published**, since the author of a course can hide it from the users at first, and only make it visible once the course has the base content in place. Generally speaking, many Udemy courses (especially the more popular ones) are constantly changing and this study has made ample use of this fact, but this statistic is still useful, if slightly less rigorous and more intuitive than desired. Figure 4.6 shows a histogram of the time between course creation and publication (in years). The median time is about 9.5 days, while the 3rd quartile is about 36 days. This means that 75% of courses are published at most slightly more than a month after their creation. Of course, this value is not entirely reliable, since it is possible for a creator to create the content for their course beforehand, then create it and quickly publish it. Nevertheless, the fact that the majority of courses have such short creation to publication times shows that most of these courses do not take much time to create, contributing to the agility of the platform. Looking at the length of these courses helps reinforce the idea that these courses are not very time-consuming to create: the median length of a course is 3.5 hours, while the third quartile of course length is 6.5 hours. The sharp focus of many Udemy courses on small topics that can be taught in a short amount of time is therefore a contributing factor to the relatively short times that they take to make.

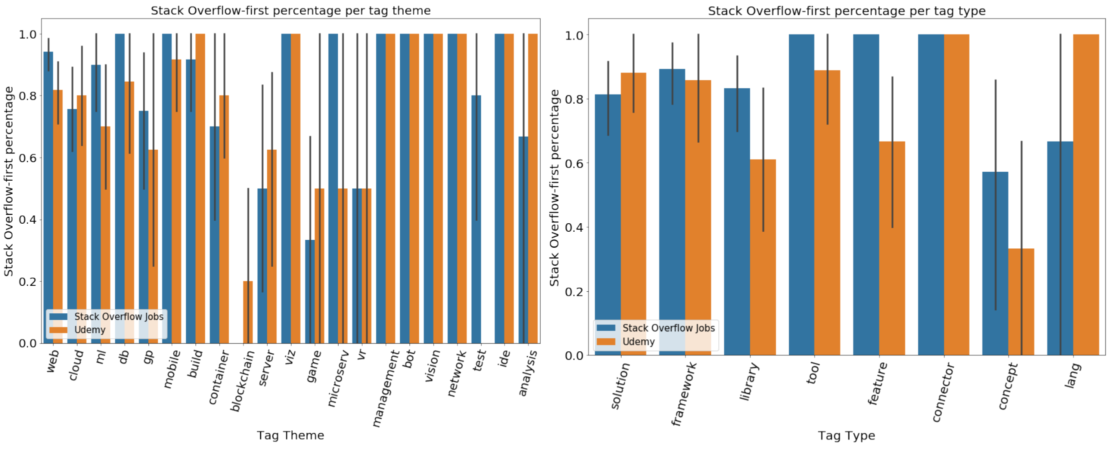
Stronger signals of rising importance

Tables 4.6 and 4.7 show two things: 1) Stack Overflow Jobs is relatively agile (with a median delay of 119 between the first question and the first ad, for "all" tags), and 2) Udemy has become considerably more agile over the years. This means that for predicting the appearance of a tag on either Udemy or Stack Overflow Jobs, the prediction window available is 3 to 4 months. As a result, the predictive aim now begs an important question: what other signals are available get from Stack Overflow before the appearance of the tag on Udemy or Stack Overflow Jobs, and how strong are they? In order to answer this question, two sets of measures have been calculated for each of the two expert/group-driven platforms (the plots have been moved to the appendices for brevity's sake):

1. For each tag, the delay between the N-th Stack Overflow question and the first appearance on the expert/group-driven platform (for "all" tags, post-October 2016), for various values of N.
2. For each tag, the delay between the first N-vote week on Stack Overflow[12] and the first appearance on the expert/group-driven platform (for "all" tags, post-October 2016), for various values of N.

Based on the values of the aforementioned two sets of measures (which can be seen in the box plots found in Appendix C), the (median-case) lead that the first Stack Overflow question and the first 1-vote week have on MOOCs and ads, quickly shrinks or is even reversed as N is increased. In particular, in case of votes, this lead is quickly reversed and becomes negative: for example, the median delay between the first 5-vote week of a tag and its first Udemy mention is -55 days, meaning that Udemy precedes the first 5-vote week by almost 2 months! This makes it increasingly clear that our predictive aim may be very difficult to achieve (if not downright infeasible): even these early and weak signals - which also exist for many tags that *do not* appear on Udemy/Stack Overflow Jobs - usually tend to come very shortly before (or even *after*) the appearance of the tag on the expert/group-driven platforms, and thus leave only a very small prediction window.

[12] The first week where all of the tag's questions combined get at least N votes.



(a) (b)

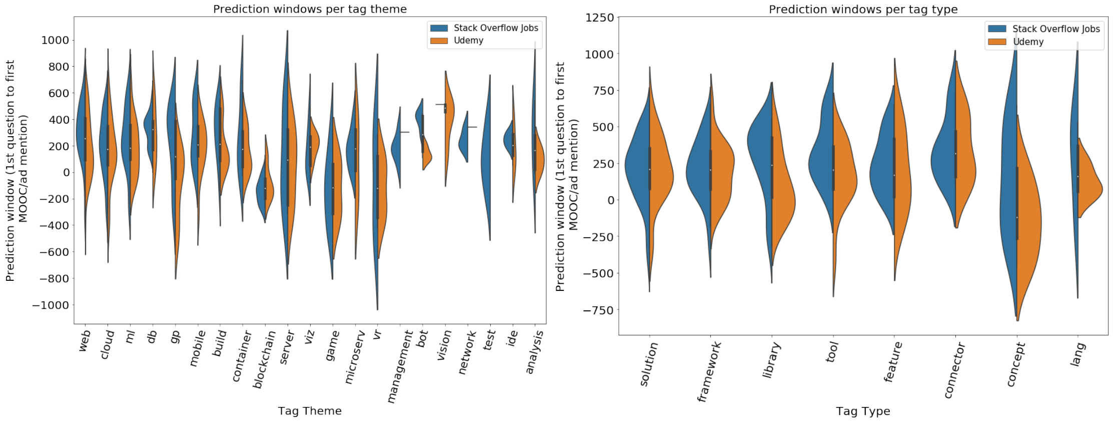
Figure 4.7 – Barcharts (with error bars) of the proportion (from 0 to 1) of Stack Overflow-first tags in every (a) tag theme and (b) tag type. The orange bars are for tags matched between Stack Overflow and Udemy, while the blue bars are for tags matched between Stack Overflow Jobs and Stack Overflow (and therefore, tags that appear on both Udemy and Stack Overflow Jobs are counted in both bars).

4.4.3 Tag themes and types

As mentioned before, this study is not complete without an analysis of the **content** of the tags that have been studied. In order to investigate the differences in appearance orderings and prediction windows between various tags, the first step was to choose a set of tags, and manually annotate them with their "themes" and "types". Tag themes indicate the general subject area of the tag, including web, cloud, machine learning, general-purpose application development, etc. A tag's type is an indicator of its granularity: some are about concepts, some are full-fledged technological solutions, some are development frameworks, while others are libraries or features in a larger language or framework. Due to the difficulty and sheer scale of this annotation task, a subset of the tags were chosen for it: the "popular" post-October 2016 tags that appeared either on Stack Overflow Jobs *or* on Udemy. This results in a total of 227 tags. The annotation of each tag was performed by looking at its Stack Overflow description excerpt and its online documentation, using the definition of that concept or technology to annotate it with one type and at least one theme. The definitions of the tags themes and types can be found in the table in Appendix D.

Two sets of metrics are of interest here:

1. The proportion (from 0 to 1) of Stack Overflow-first tags among tags of each theme and



(a)

(b)

Figure 4.8 – Violin plots of the prediction windows (in days) for tags in every (a) tag theme and (b) tag type. Orange is for tags matched between Stack Overflow and Udemy, while blue is for tags matched between Stack Overflow Jobs and Stack Overflow. Again, tags that appear on both Udemy and Stack Overflow Jobs are counted in both.

of each type, and

2. The distribution of the prediction windows (i.e. time to first appearance on Udemy/Stack Overflow Jobs) among tags of each theme and of each type.

Fig. 4.7 shows bar charts of the former (with error bars), while Fig. 4.8 show a box plot of the latter. Both figures show the statistics separately for tags matched to Stack Overflow Jobs and for those matched to Udemy, given the clear differences between the two. Generally, as can be seen in the two figures, the different types and themes are not so strongly differentiated in terms of either the Stack Overflow-first proportion or the prediction window, but there are exceptions:

- Among tag **themes**, themes such as “blockchain”, “game”, and “server” seem to have much lower Stack Overflow-first proportion (and they are also less popular, see Appendix D), and the former two also seem to have much smaller prediction windows with negative medians. Statistically speaking, this could be attributed to the smaller numbers of tags for these themes, but the fact that there are fewer new tags for these topics could also indicate less interest for these themes on Stack Overflow, which is reasonable as Stack Overflow is more about coding-related questions. On the other hand, “web” is both by far the most popular in terms of the number of tags, and also has some of the highest Stack Overflow-first proportions with relatively low variance. Themes like “db” (databases), “mobile” (mobile app development), and “build” (code build tools) are, although less popular in terms of raw tag counts, are also some of the most reliably Stack Overflow-first themes. Themes such as “cloud” (cloud computing) and “ml” (machine learning) have slightly lower Stack Overflow-first proportions, but it is not clear why that is the case[13].

- Among tag **types**, “solution”, “framework”, and “tool” are generally reliably Stack Overflow-first. The “library” type has a slightly lower Stack Overflow-first percentage, while “concept” has both a much lower proportion and a much smaller median prediction window. The former could be attributed to the greater specificity of libraries compared to the larger, more coarse-grained solutions and frameworks. The latter could, again, be attributed to how Stack Overflow is mainly a programming Q&A platform and non-coding-related questions, such as those asking about concepts per se, are not its main focus.

So, to summarize, the insights gained from analyzing the types and themes are mainly that Stack Overflow tends to be quicker when it comes to relatively general coding-related tags, while tags on more niche programming topics (e.g. individual libraries) and less coding-related tags tend to appear on Stack Overflow with greater latency.

4.5 Discussion

4.5.1 Implications of results

Implications of descriptive findings

The descriptive aspect of this study serves the purpose of establishing a broad (while reasonably deep) view of an entire professional domain, enabling an understanding of its broader dynamics when it comes to new topics. It is designed not for zooming in on individual or small groups of innovations, but rather for surveying all the emerging topics in that domain, with its main focus being on understanding the *platforms* involved, rather than the topics per se. The descriptive findings of this study have quantified the agility of the software programming domain. In addition, the results show that the platforms that are more novice/individual-driven can provide earlier information about the appearance of new topics, but given their relative lack of expert curation (compared to platforms like Udemy and Stack Overflow Jobs), the signal from them comes with considerable noise, in the form of new tags that are not really important new topics and do not end up being used more than a handful of times. This trade-off between earlier but noisier information versus later but higher quality information is

[13] It is worth mentioning, however, that more theoretical and non-programming machine learning questions can be asked on one of Stack Overflow’s sister websites, Cross Validated, therefore making machine learning one of the themes for which Stack Overflow is not the ultimate Q&A website.

intuitive, but the details found in the study, such as the greater agility of Udemy compared to Stack Overflow Jobs, and the variability of tag agility based on their themes and types, allow for the suggestion of a prioritization of platforms for people who need to understand emerging skill trends. The suggestion would be to first look at Stack Overflow's new tags, taking into account the semantics of those tags (i.e. the theme, type, and software versions if applicable), and then to look at Udemy to compare the trends on the two platforms and make use of the slightly less agile but expert-created and higher-quality insights that can be gained from Udemy. The reason Udemy is placed second, ahead of Stack Overflow Jobs, is twofold: firstly, Udemy has greater agility compared to Stack Overflow Jobs, and secondly, many of the tags that appear on Stack Overflow Jobs are not considered important by the large and diverse Stack Overflow community. These two reasons mean that Udemy complements Stack Overflow better than Stack Overflow Jobs does.

Implications of prediction-related findings

Given the results for the predictive aim, the findings can be summarized as follows:

- The appearance orderings observed for all tags show that for as many as 29% of the tags in Udemy's case and 35% of the tags in Stack Overflow Jobs' case, the prediction task is meaningless, since those tags have appeared on the expert/group-driven platforms before appearing on Stack Overflow.
- Google Trends is not useful for this prediction task.
- The prediction windows calculated for recent (post-October 2016) tags show that the window for the median tag is about 3 months for Udemy and about 4 months for Stack Overflow Jobs. Given that it is time-consuming to create even a single lecture (let alone an entire course), this is a small time window and is a testament to the agility of the software ecosystem.
- Udemy and Stack Overflow Jobs prediction windows for stronger signals, such as the n-th Stack Overflow question (n from 2 to 10) or the first n-vote week on Stack Overflow (n from 1 to 10) are ever smaller. The medians approach one month in case of question-related signals, and go into negative values in case of vote-related signals. The signals, whose prediction windows have been calculated, are signals that are also present for many of the tags that do not appear on any expert/group-driven platform.

The conclusion from these is that the predictive task, using user activity data from Stack Overflow and Google Trends, aiming to predict the appearance of a lecture/course or ad on each tag, is unlikely to be successful. This is due to the cap on the number of tags it is meaningful for, the scarcity of useful features in the data, and the small time window available for the prediction in most cases. Given Stack Overflow's popularity, and the fact that creating a question on Stack Overflow intuitively takes much less effort than creating a course lecture or a job ad, this shows the agility of the software industry and how quickly job ad and MOOC platforms catch up with emerging trends.

What this study rules out, specifically, is predicting the appearance of the first course/lecture or job ad using user actvity data on Stack Overflow. The prediction of further interest in the topic (through more courses, lectures or ads) is an interesting question to explore, and may be feasible. The purpose of the prediction of the n-th course/lecture or ad, for example, could be to better understand the pace of the adoption process and the later interactions between the different platforms. In particular, the first Udemy course or lecture addressing a new tag may not be ideal – the most successful course on a subject is not always the first. Having a classifier that can predict a topic's wider discussion Udemy can allow other content creators to become aware of the importance of that tag earlier, and to potentially create better, more comprehensive lectures or courses.

In addition, the predictive task in this study was focused on features coming from *user activity*; a prediction relying on the semantics of the tags (e.g. the theme and type of the tag, whether it is a major or minor version of an existing popular technology, etc.) may be more successful. The downside to such a predictive approach would be the need to label each tag's features (which may be difficult to fully automate), and the fact that, as we showed in the results section, there would still only exist a very small window of time for predicting the *first* lecture, course, or ad.

4.5.2 Generalizability

An important question when it comes to the methodology presented here is the degree to which it could be applied to domains other than software programming. There are two aspects to this discussion: whether such a study would be *possible* for another domain, and whether it would be *appropriate*.

Regarding the possibility, this methodology is extensible to other domains as long as 1) online hiring and educational platforms exist for that domain, and 2) there is a way to detect those new topics in that domain. In this study, Stack Overflow tags served the latter purpose. Stack Overflow is part of a family of websites called Stack Exchange that are Q&A platforms for various subjects (such as the English language, mathematics, machine learning, etc.), meaning that the detection of new topics is a non-issue for domains with a Stack Exchange website dedicated to them (since they all have a tag system similar to Stack Overflow's), and these domains also have a Q&A data source. For other domains where there is no relevant Stack

Exchange website or the relevant website is not sufficiently popular, things get complicated: the new topics have to be found using another source, and the benefits of tags (i.e. being crowdsourced and continuously updated) may be lost. The other prerequisite, i.e. job ads and other educational data, is easier to obtain: there are massive online job ad collections available (e.g. Burning Glass Technologies), and MOOC platforms like Udemy offer courses on a wide variety of topics.

Regarding the appropriateness, software programming is a domain where many, if not most skills can be self-taught, with the 2019 annual Stack Overflow Developer Survey[14] showing that over 85% of respondents report having taught themselves a new language, framework, or tool without a formal course, while over 60% have taken a MOOC. Therefore, informal learning is feasible and quite widespread in this domain, making online Q&A platforms and skill-based MOOC platforms ideal for software developers. This contributes to the agility of these platforms for this domain, making them prime targets for monitoring when it comes to detecting new and emerging topics. Other domains' skills, however, may not lend themselves so much to such learning methods. In particular, online Q&A may fail to be as agile in domains where there are fewer experts, where digital technology has achieved less penetration, or where a lot of knowledge may be proprietary, e.g. large Q&A forums on operating industrial machinery are less likely. MOOCs may also take up more secondary roles in such domains, especially in those where tangible objects play a greater role, making in-person learning more necessary.

4.5.3 Other MOOC platforms

The MOOC platform chosen for this study was Udemy due to the fact that anybody can create one, and the MOOCs are skill-based and self-paced. However, it is also important to investigate the differences between various MOOC platforms and their relative agility, given the fact that most platforms other than Udemy offer courses by institutions, rather than individuals. Most such MOOC platforms (e.g. Coursera, edX) have academic or quasi-academic schedules for their courses, which means that students can only take courses during certain specific periods of time, although within that time period, the student can progress through the course at their own pace (up to the end of that time period). This means that courses are available only a limited number of times per year. For example, according to the edX website, edX courses usually run at least once a year, and courses that are part of a program run at least three times a year (which would be once every four months). The results of this study showed that in many cases, Udemy courses manifest a new topic less than 4 months after it appears on Stack Overflow. Since these courses can be taken at any time, Udemy has an undeniable edge over platforms that have academic schedules when it comes to agility, because even if an edX

[14] https://insights.stackoverflow.com/survey/2019

course is always updated with the latest topics, students would not necessarily be able to take the course immediately, thus potentially increasing the delay between the time when the topic first appeared and the time when students are first able to study that topic.

Regarding a more in-depth analysis of edX and Coursera vis a vis Udemy, a request was made to obtain their data to conduct an in-depth analysis (as the necessary data, such as course creation dates, are not available on their respective websites), but it went unanswered. A very small set of edX data is available from Kaggle[15], but the analysis of the creation dates of edX computer science courses versus their closest Udemy counterparts was inconclusive due to the minuscule volume of data (with only 13 unique courses), which prevented any statistically significant analysis.

4.5.4 Threats to validity

In the methodology section, several caveats arising from methodology design choices were discussed. This necessitates a discussion of how those caveats can pose a threat to the validity of the study's results:

- There are some caveats to how representative Stack Overflow and Google Trends are of non-expert behavior, because not every person with a question will ask it on Stack Overflow, and not everyone performing a search is a non-expert. However, given the popularity of Stack Overflow and the prevalence of Google as a search engine (and the fact that there are, generally, many more non-experts than experts), the impact of these caveats is expected to be limited, and the classification of these two as more novice-driven compared to Udemy and Stack Overflow Jobs is in any case accurate.
- Only a small subset of the tags get matched between all our datasets. However, the separate analysis of Stack Overflow with each of the expert/group-driven datasets and the similarity of the results for these different dataset goups means that this does not present a considerable threat to the findings of this study.
- In general, some tag matches may have been missed, given the use of exact matching in the syllabus of a Udemy course or in the description of a job ad. As a result, the matching most likely lacks perfect recall on those datasets. The tag matching could also have imperfect precision, as sometimes an n-gram might erroneously match with an existing tag, although intuitively, this should be less likely.
- In the methodology section, two potential issues were mentioned with respect to tags: that a tag might not represent a new or important topic, and that a new topic might

[15] https://www.kaggle.com/edx/course-study

never appear as a tag. As discussed back then, the latter should be a minor issue as Stack Overflow is a very popular Q&A website, and the former has been addressed by analyzing popular tags separately.

- This study's criterion for the appearance of a tag in a course is its appearance either in the title of the course, or in the title of one of its lectures. This means that the tag in question does not necessarily have an entire course dedicated to it, which also goes the other way around: not every new topic *needs* an entire course. Training program decision makers wanting to use a methodology like ours to track new topics in a domain should be mindful of this unevenness in granularity.

- Since Stack Overflow is where all of the tags come from, there is a certain degree of bias towards the types of topics that end up being tags on Stack Overflow, or rather against those that do not. This could have an effect on the assessment of the appearance orderings. However, this is more likely for Udemy, rather than Stack Overflow Jobs, because the latter and Stack Overflow share tags, and a tag can be created on Stack Overflow Jobs rather than on Stack Overflow. This bias is most likely to affect softer and less coding-related skills.

4.5.5 Future work

The present work lends itself to multiple directions for future work. The most straightforward direction is to apply the methodology to another professional domain, which would also allow for a comparison of different domains. A second direction is to analyze the **spread** of new topics on the platforms in this study, going beyond the first appearance. This would be an analysis of the popularity of various new topics over time on the different platforms. Clustering these new topics together (similar to what we did in this paper with the tag themes and types) would then reveal broader trends in the professional domain, enabling a deeper understanding of the trending new topics (which would be of interest to training program creators), and it would also open the door to alternative prediction tasks, e.g. predicting the popularity of topics on the expert/group-driven platforms based on their early behavior on *all* the platforms (and not just the novice/individual-driven platforms). This future direction is quite extensive and could form the basis for multiple studies. Finally, another natural direction for the extension of this work is to include other, less open and more centralized MOOC platforms, and even more traditional educational instutitions such as universities in the study, which were not included in this study in order to keep the scope reasonable and because Udemy, with its unique properties, was a very interesting platform to study.

4.6 Conclusions

In this chapter, a methodology was proposed for analyzing the dynamics of new topics among online educational platforms and hiring platforms in the software programming domain. The results show that novice/individual-driven platforms such as Q&A websites, where content creation is often initiated by novices and the content-creation decisions are made individually, are generally faster at manifesting emerging topics compared to educational platforms and corporations. The impressive agility of the software programming domain was quantified, demonstrating that it can be indeed very difficult to predict the digital traces of the earliest adopters of a new skill or technology. This work is a first step towards understanding the relationship of these platforms with each other, and it has two main implications for training program creators in the software programming domain: first, that Stack Overflow is a largely reliable data source for tracking emergent topics; second, that given the agility of the software programming domain, its MOOC and job ads platforms, especially Udemy, can also provide early signals on these emerging topics. In accordance with these implications, the present methodology allows course creators and training experts to gain insights into how quickly each sub-domains of the software programming domain is evolving and which platforms are quicker at manifesting the changes, therefore allowing them to prioritize their attention and resources on the most pertinent sub-domains and to focus their further analyses on the right platforms.

5 Study 3: Emerging Skills

5.1 Introduction

As discussed in the previous chapters, the present era is full of disruptive forces acting on labor markets. The AI and automation-centric Fourth Industrial Revolution (Brynjolfsson and McAfee, 2011, 2014; Maisiri et al., 2019) and the COVID-19 pandemic (Agrawal et al., 2020) have quickly made some skills outdated while raising others to prominence. These forces introduce rapid change into the skill sets required for many jobs and domains, and the speed at which these changes happen complicates the process of curricular change, as organisational cycles might struggle to keep up with the pace (Ellis, 2003; Brynjolfsson and McAfee, 2011). Identifying the important skills and predicting the important skills of the future is thus a crucial task that could help training providers stay on top of the trends.

However, as noted in the literature, the prediction of important future skills is a very challenging task (ILO and OECD, 2018), and the results of the predictive part of Study 2 are a testament to its difficulty. Since predicting the first appearance of skills on job ads is out of the question, it is reasonable to instead work on the early identification of fine-grained "emerging skills", i.e. skills that are rising to importance from relative obscurity. Since existing skill trend analysis methods often investigate coarse-grained skills rather than fine-grained ones (Strack et al., 2020; CEDEFOP, 2018; Boehm, 2006) and many approaches focus on describing the present rather than predicting the future (Strack et al., 2020; Szabó and Neusch, 2015), this is a gap that needs to be filled.

Defining "emerging skills" as **previously low-demand skills that have recently experienced a surge in hiring demand**, this study presents a classification pipeline with the aim of predicting the emerging skills of the near future. In other words, the goal is to predict the surge in hiring demand before it occurs. The driving hypothesis behind this study is that the job ad time series of each skill, indicating demand for the skill over time, contains signals that help predict whether or not it is going to emerge in the near future. Applying this methodology to data from the ICT sector in Singapore, such a predictive task is found to be feasible, confirming that job ads contain information that can be used to predict emerging skills. Then, the strengths and weaknesses of the classifier models are investigated, and the distinguishing signals of emerging skills are identified, concluding that non-linear growth and spikes are the most important features of an emerging skill's job ad time series.

This chapter will first discuss the existing literature on TNA and skill trend analysis with a focus on analyses of online datasets, touching upon both the literature first introduced in Chapter 2 and the literature specific to the problem at hand. This way, the gaps in the literature on the relevant literature are laid out. Then, the research questions and methodology are laid out. Afterwards, the results of the classification pipeline are presented, answering the research questions based on those results. In the end, the implications and limitations of the present work are discussed and several directions are proposed for future work, aimed at rectifying the limitations of this methodology and improving its ability to predict the emergence of skills. The contents of this chapter are taken from the paper "On the radar: Predicting near-future surges in skills' hiring demand to provide early warning to educators", which we published in the journal "Computers and Education: Artificial Intelligence".

5.2 Related work

5.2.1 The necessity of curricular change in ICT

The disruptive effects that automation, AI, and other disruptors such as the COVID-19 pandemic have had on many industries cannot be overstated, and this is particularly the case in the ICT domain and those related to it. For example, the COVID-19 pandemic brought about a sudden switch to teleworking, which in turn caused a surge in the need for basic digital skills, such as the use of teleconferencing software (Agrawal et al., 2020). The disruption brought about by AI and automation is even more fundamental, as many tasks that were previously only feasibly done by humans become doable by increasingly intelligent machines (Illanes et al., 2018). This covers a wide range of tasks, from driving a vehicle, delivering goods to customer care, even diagnosing disease (Forbes, 2019). At the same time, there is an explosion in the demand for skills relevant to the new industry, such as technological, programming, and data analysis skills (Goldfarb et al., 2021; Maisiri et al., 2019). All of these changes are happening in a short time frame, and research shows that many institutions, including educational institutions, have fallen behind (Brynjolfsson and McAfee, 2011). This makes developing new analytic and predictive methods for the skills landscape of the ICT domain absolutely imperative, and such methods would be of great use for other domains as well.

5.2.2 Big data and skill landscapes

There significant body of methodologies for curricular change and training needs analysis were explored in Chapter 2, with big data from online labor market intermediaries emerging as the one opening new avenues for work on the technologies introduced and the domains transformed by Industry 4.0(Horton and Tambe, 2015; ILO and OECD, 2018).

As mentioned before, previous labor market research on skills using novel big data sources has often focused on higher-level skill or job trends (Gallivan et al., 2004; Lee and Mirchandani, 2010; Matsuda et al., 2019; Gurcan and Cagiltay, 2019), and the potential of such data for curricular change remains mostly untapped. Most of the previous works on curricular change either use expert, graduate, or student surveys to effect it (Carnegie and Crane, 2019; Fowler et al., 2014; Stevens et al., 2011), or focus on personalizing education using student learning analytics (Williamson, 2017; Cen et al., 2015). The previous work that is of particular interest to this study are analyses of more granular skill trends (Strack et al., 2020; BGT, 2019; Dawson et al., 2019). Some of these are conducted by the corporations that host or own the data, while others are academic research. For example, the whitepaper published by the Boston Consulting Group and Burning Glass Technologies in 2019 (Strack et al., 2020) groups skills into five categories, based on two factors: their overall hiring demand, and the growth of this demand. One of these categories, which they call "high-growth skills", is the main inspiration for our work. These are defined as skills with fewer than 10,000 ads in three years, whose growth over these years has been over 40%. These are the skills that are growing rapidly, but which are less likely to have already been identified as important due to their low previous popularity (compared to those skills that are growing fast *and* already enjoyed significant popularity to begin with). This study's concept of "emerging" skills is essentially a generalization of this concept, without the specific thresholds. Combining this idea with the skill demand projections common in the literature is the basic idea behind the present study. Another interesting approach to the problem is found in (Dawson et al., 2019), where the authors use several hand-picked measures — including the growth in demand for a job title and its predictability — for detecting high-level skill shortages in Australian job ads. Their work particularly touches upon the difficulty of predicting hiring demand, although in their case, it is for job titles rather than skills. The work in (Dawson et al., 2020) uses a similar approach. In both cases, the authors find that job ads and employment statistics are the most predictive features when it comes to yearly skill shortage changes. However, previous work has never focused on projections of growth for fine-grained and less popular skills, and that is the gap this study aims to fill.

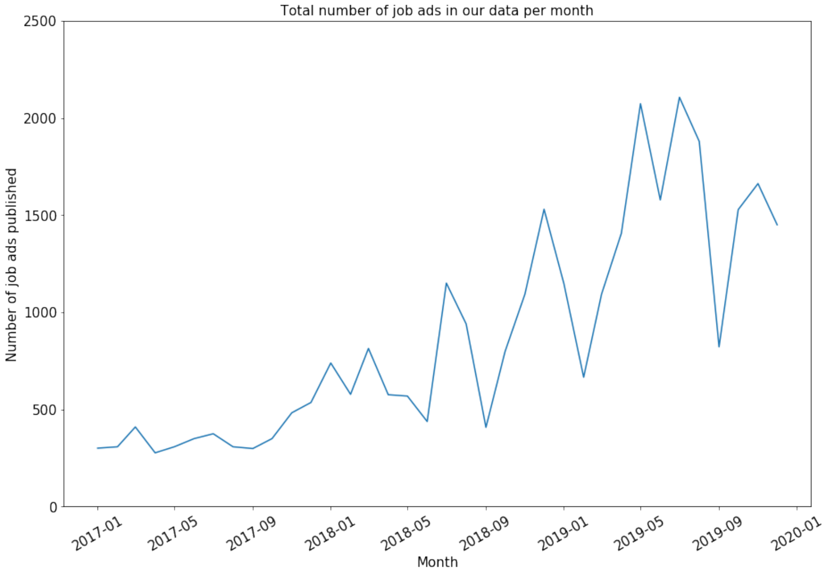


Figure 5.1 – Number of job ads per calendar month in the 2017–2020 period in the data. Note the rather drastic growth of the number of ads over time, which may be due to a growth in the popularity of JobTech itself. Considerable drops in job ad counts can be observed both in late summer and around the time of the Chinese new year in the later years.

5.3 Objectives and methodology

5.3.1 Data and definitions

The data in this study consists of all the job ads in the Singaporean Information and Communication Technology (ICT) sector between the beginning of 2017 to the beginning of Q2 2020[1], although only the data between the beginning of 2017 and the beginning of 2020 are examined. This is done in order to exclude the disruptions caused by the COVID-19 pandemic in 2020 (since the effects of the pandemic are not a focal point of this study). Every job ad in the dataset contains the company posting the ad, the date the ad was posted, the textual description of the ad and skills extracted from it. For the chosen period (2017–2020), the dataset contains a total of 31,350 job ads, spread across 2,264 companies and involving 987 skills that can in some way be called ICT skills. These skills include both programming-related skills and skills related to using specific computer software (such as Microsoft Office products, Adobe products, etc.). Figure 5.1 below shows the total number of job ads in the dataset on a monthly basis.

The analysis in the present study relies on the job ad time series of each skill in order to predict the skills that will have a surge in hiring demand in the near future. However, looking at the

[1] The job ads come from JobTech, a Singaporean online hiring platform who have kindly provided the data through SkillsFuture Singapore (SSG) as an intermediary. All data rights belong to JobTech.

number of ads that include the skill in a particular period of time (e.g. a month) is only one way to analyse the trends of that skill in job ads. In order to formalize this point, let us define two concepts. The **hiring volume** of a skill is the number of job positions that have been announced for it in a particular time period. The **hiring spread** of a skill is the number of companies that have announced job positions for a skill in a particular period of time. Based on these two concepts, let us introduce three types of job ad time series for skills. These will serve two purposes: they will allow for a precise definition of emerging skills, and will serve as competing data inputs to the classification pipeline.

1. Raw popularity (rawpop): The value of the skill's time series for each period of time t (whose length can be one month, one quarter, etc.) is simply the total number of job ads posted for it during that period:

   $$rawpop_{s,t} = \sum_{c \in companies} ads_{c,t}$$

   Where $ads_{c,t}$ is the number of ads posted during time period t by company c. This type of popularity (and by extension, time series) ignores hiring spread and only emphasises hiring volume.

2. Logarithmic popularity (logpop): For the value of the skill's time series for period t, instead of summing up the total number of ads each company has posted for the skill, the logarithm of that number is computed and then summed up:

   $$logpop_{s,t} = \sum_{c \in companies} log(1 + ads_{c,t})$$

   What this type of popularity does is strike a balance between hiring volume and spread: one more ad by a company that has already posted an ad for the skill is worth less than an ad by a company that has not already posted an ad for it. However, it does not throw hiring volume out the window entirely, as more ads by the same company still matter, albeit less than they would in rawpop.

3. Binarised popularity (binpop): The value of the skill's time series for time period t is simply the number of companies that have posted an ad for it:

   $$binpop_{s,t} = \sum_{c \in companies} I_{\{x>0\}}(x = ads_{c,t})$$

   Where $I_{\{x>0\}}$ is the indicator function that is 1 for positive numbers. This type of popularity throws hiring volume out entirely and only focuses on spread.

An example demonstrating the differences between the three popularity types can be seen in

		Skill 1	**Skill 2**	**Skill 3**
# of ads by companies	**Company 1**	1	2	3
	Company 2	1	2	3
	Company 3	1	0	3
	Company 4	1	10	3
Popularity type	**Rawpop**	4	14	12
	Logpop	2.77	4.60	5.55
	Binpop	4	3	4

Table 5.1 – A comparison of the three popularity types for a hypothetical example. For example, Skill 2 has 2 ads posted by Company 1 and 2 each, and 10 ads by Company 4. Although the rawpop of Skill 2 is the highest, recruitment for it is mostly concentrated in Company 4, resulting in its binpop being lower than that of skills 1 and 3. In addition, the logpop of Skill 3 is again higher than Skill 2's due to greater spread, despite Skill 2's higher rawpop.

Table 5.1.

Ground truth and data points

The last preliminary to cover before describing the classification pipeline is to discuss the ground truth that is going to be predicted. The definition given for emerging skills so far is a rather vague one, and needs to be specified further for the prediction task. The two vague parts that need to be specified are "recency" and the "size of the surge".

Let us denote the rawpop of a skill s in the year y by $Pop_{s,y}$, and the n-th quantile of a set T as $Quantile(T,n)$. Also, let us define the quantities $Prevpop_{s,y}$ and $Growth_{s,y}$ as follows:

$$Prevpop_{s,y} = Pop_{s,y-1}$$

$$Growth_{s,y} = Pop_{s,y} - Prevpops, y - 1$$

Then, s is declared to be an emerging skill in the year y if it satisfies:

$$Growth_{s,y} >= Quantile(\{Growths, y\}_{s \in Skills}, q_L)$$

$$Prevpop_{s,y} <= Quantile(\{Prevpop_{s,y}\}_{s \in Skills}, q_U)$$

where q_U and q_L are, respectively, quantile upper and lower bounds on previous year popularity and popularity growth. The first condition (with the quantile q_L) requires the skill to have grown considerably, and eliminates skills that have not experienced a surge in hiring demand from one year to the next. However, with q_L alone, the result would be growing skills, rather than emerging skills. This is why the second condition exists (with the threshold q_U): putting an upper bound on the previous rawpop of a skill enforces the recency part of the definition, as the skill must not have already been too popular in the year $y-1$. Since q_U and q_L are quantiles, they allow the upper and lower bound values to be determined from the data itself, and for simplicity's sake, the same q_U and q_L pair will be used for all years. These two quantiles are two degrees of freedom in the model, and they decide the general popularity level and growth of the skills that are deemed emerging. For example, lowering q_U will push some of the more popular skills into the non-emerging set, while increasing q_L will shrink the emerging set by making sure that only skills with larger growth values are deemed emerging. It is not a given that the classifier model would work well for any choice of q_U and q_L (and as the results will show, it does not), and later sections will discuss how their values can be set.

It is worth discussing here that the concept of emerging skills does not *have to* be defined through pure hiring volume (i.e. rawpop); it could also be defined based on hiring spread. Such a definition would focus on how much the skill has spread among companies, rather than how much hiring has happened for it. However, for this study's main objective of providing insights to training providers on which skills are more in need of training programs, the number of available positions for a skill (which is equivalent to raw demand) is of much greater importance than its spread among companies. As such, the specific definition of emerging skills will be based on hiring volume, rather than spread. However, the question of whether or not signals from hiring spread can help predict hiring demand is a different one; this question will be explored by pitting the three previously-defined popularity types (rawpop, logpop, and binpop) against each other as competing inputs to the predictive pipeline, and comparing the performance of their respective models.

It is worth noting that a caveat of the aforementioned method for generating ground truth is that some skills can emerge in successive years: a skill s could be below the threshold q_U for both the year $y-1$ and the year y and be above the threshold q_L for both years, thus putting it into the set of emerging skills twice in a row. The implications of this situation will be discussed in the results section.

Once the ground truth has been generated, for each type of job ad time series (i.e., each popularity type), data points can be created, which will form the basis of the training and test sets. These data points will be called **skill–periods**, with a skill-period for the year y consisting of the skill's job ad time series for the entirety of year $y-1$, along with the ground truth label of the skill in the year y (1 if emerging, 0 if not).

The full training set consists of all the skill–periods for the year 2018, whereas the full test set consists of all the skill–periods for the year 2019. This year-based split is necessary to avoid information leaking from the test set into the training set. In order to compute confidence intervals for our performance measures, skill-based splits will be created within the year-based split, wherein each classifier is trained and evaluated on several random subsamples of the full training and test sets, respectively. In each such subsample, some skills are randomly selected to be in the test set, and are removed from the training set, thus making the training and test sets disjoint both in years and skills. This helps ensure that there is no information spillover from the training set into the test set. A total of 20 skill-based splits will be used to create confidence intervals for the precision, recall, and F1 scores of the trained classifiers.

5.3.2 Classification pipeline

Extracting features

The input to the classifier consists of features extracted from time series (where the time series come from the skill-periods). The features extracted include summary statistics (e.g., mean, various quantiles, variance), linear trends, measures of non-linearity and spikes, FFT coefficients, and many more[2]. Many of these features are intuitively expected to be important (e.g., fast linear or non-linear growth or large spikes can be indicators of quick "emergence"), and the completeness of the set of features ensures that no potentially useful signal is missed out on.

Before feature extraction, each time point of each skill–period is median-normalized using the median of that time point. Then, moving average smoothing is applied to reduce noise in the time series. Afterwards, feature extraction is performed on all the data points.

After the feature extraction, feature reduction is necessary in order to avoid overfitting. This is because the number of data points is quite limited (around 1000 in each of the training and test sets), and the number of extracted features is relatively large (around 300). The feature reduction pipeline used here has two steps. In the first, feature selection is performed on the training data to eliminate some of the less discriminating features. This is achieved by performing a one-way ANOVA for every feature and the output, choosing the top N_1 features in terms of F-value. In the second step, Principal Component Analysis (PCA) is applied to the training data and both the training and test data are projected into the new subspace spanned by the top N_2 principal components. The values of N_1 and N_2 (the number of features after feature selection and PCA, respectively) are, along with the model's other hyperparameters, determined using cross-validation (with the F1-score as the evaluation measure).

[2] The features have been extracted using the Python package tsfresh. Its documentation, including a full list of the extracted features, can be found at https://tsfresh.readthedocs.io/en/latest/.

Classifier models

For the classifier, two competitor models are designed: a **one-step binary classifier** model that predicts the binarized ground truth directly, and a **two-step regression model** that predicts $Growth_{s,y}$ itself. The output of the binary classifier model can be evaluated directly, whereas the q_L quantile will be used to binarize the output of the regression model for evaluation (making it a *two-step* classifier). The reason why a regression model is used as the second model type is that the binary ground truth may be noisy near the $Growth$ lower bound (i.e. the difference between a skill above the threshold and a skill below it may be quite small). Since the regression model predicts $Growth$ itself, it avoids that noise entirely in its training (although the noise from the $Prevpop$ upper bound will still be present).

The **one-step binary classifier** is a logistic regression model trained on the binarized ground truth with a post-filtering step, in which its predictions are only computed for the skills with $Prevpop$ below the upper bound, while the rest are predicted as negatives. The post-filtering step essentially means that the classifier only learns the indicators of growth that appear in the time series of emerging skills. In other words, among the skills below the $Prevpop$ upper bound, it learns to discriminate between those that would grow considerably in the next year and those that would not. It does not, however, learn to discriminate between skills with $Prevpop$ values above the threshold and those with values below the threshold. This makes sense, as this threshold is always a known value, even in a real future prediction scenario (since, for example, the upper bound for 2019 skill-periods is computed using the job ad time series in 2018, and uses no information from 2019). In line with this post-filtering step, a pre-filtering step is performed as well: the skills with $Prevpop$ above the upper bound are removed from the classifier's training set. This means that the skills the classifier trains on are emerging and not-yet-emerging, while it sees none of the has-already-emerged skills.

The **two-step regression model** is a ridge regression model which, instead of training on the binarised ground truth, learns to predict $Growth$ directly. Denoting the output of the model as $PredictedGrowth_{s,y}$, the the emerging skills are those where

$$PredictedGrowth_{s,y} >= Quantile(\{PredictedGrowth_{s,y}\}_{s \in Skills}, q_L)$$

$$Prevpop_{s,y} <= Quantile(\{Prevpop_{s,y}\}_{s \in Skills}, q_U)$$

and all the rest are predicted as non-emerging. The same pre-filtering step is applied, involving the deletion of skills above the $Prevpop$ upper bound from the training set. Much like the logistic regression model described above, this model learns the signals of growth in emerging skills. However, it has three potential advantages over the binary classifier model. First of all, as discussed before, it could potentially avoid the noise introduced by the binarized ground truth. Secondly, the direct forecasting of each skill's $Growth$ can be useful *per se*. Finally, the forecasting of growth in demand means that the model could also be used to predict the skills that are expected to decline in popularity (although this study is not concerned with such a prediction).

Baselines

In order to evaluate the performance of the classifier models, baselines are needed to serve as points of comparison. The structure of the classification problem lends itself to several types of baseline:

1. **Previous-year baseline**: This baseline reports the emerging skills of the previous year as positives and the rest as negatives. This corresponds to the idea that every skill that was emerging last year will be emerging again this year (which is made possible due to the fact that skills *can* be emerging two years in a row).

2. **Below Upper Bound baseline**: This improved baseline relies on the fact that many emerging skills already have some degree of popularity before they emerge. Using the terminology introduced before, it relies on the fact that the most likely candidates for emergence in year y are the ones whose $Prevpop$ is just below the upper bound. It reports the top K most previously popular skills as emerging (and the rest as non-emerging). This is a realistic baseline, since it only relies on our past knowledge of a skill. The Below Upper Bound baseline requires training, as we need to choose the value of K; i.e., how many of the below-threshold skills we want to report as emerging. This is done using a grid search for the best value of K on the training set, using the F1 score as the evaluation measure.

5.3.3 Research questions

With the definitions and methodology all laid out in detail, the research questions can be specified as follows:

RQ1: How well can this methodology predict the emerging skills of the near future?

RQ2: How much does performance degrade when predicting further into the future?

RQ3: To what degree does the ability to predict emerging skills depend on how they are precisely defined (i.e. based on the upper and lower bounds used to specify the skills that are emerging)? Are there areas where predictive performance is systematically worse?

RQ4: What are the features of a skill's job ad time series that indicate its near-future emergence?

5.4 Results

In order to answer the research questions of this study, the models (each of which uses one particular popularity type and one of the two classifier types) have been tested against the two baselines (Previous-year and Below Upper Bound) for three different (q_U , q_L) pairs. To choose the three (q_U , q_L) pairs, a list of skills was compiled consisting of those that came up as emerging/high-growth in the reviewed skill analysis whitepapers, sorted by popularity. The parameters were then set in the three pairs such that the three corresponding emerging skill sets would cover different segments of this list. This helps ensure that the ground truth skills are reasonable, and the performance of the methodology can be tested for different (popularity-wise) specifications of emerging skills.

5.4.1 Predictive performance

The first research question of the present study is concerned with the predictive performance of the classifier model. Table 5.2 shows 90% confidence intervals for the performance of the best models for each of the three threshold sets, along with the performance of their respective baselines[3]. The three numbers in the parentheses are the 5th percentile, the median, and the 95th percentile, respectively. The first and most important takeaway from this table is that **the right classifier model outperforms both baselines for all three threshold sets**, with the gap between the median F1 scores of the best classifiers and the best baseline (which is the Below Upper Bound baseline) always being greater than 0.05.

Most of the trained classifier models soundly beat the Previous-year baseline, proving that they learn more than to simply predict all the emerging skills of the previous year as the emerging skills of the next. When it comes to the Below Upper Bound baseline, they almost always have greater precision and lower recall. However, for the purpose of this study, the results of these models are much more useful than those of the Below Upper Bound baseline, as will be demonstrated with an example. Let us take the Logpop + two-step classifier for the $(0.8, 0.65)$ threshold set, whose confidence intervals versus those of the baselines can be seen in the boxplots of Fig. 5.2. Let us train it on the full training set, and call it the **reference classifier**. This model, which beats the ensemble Below Upper Bound baseline (F1 of 0.645 vs 0.600), predicts a total of 243 skills as emerging, whereas said baseline predicts 310 as emerging (some examples of these skills can be seen in Table 5.3). The baseline has 11 more true positives (and thus 11 fewer false negatives) than the reference classifier, at the cost of 56 more false positives. Since the emerging skills predicted by the trained classifier are to be reviewed by experts, it is desirable to keep the number of false positives low, as they make experts' job harder. The

[3] Other baselines, such as simplified versions of the proposed classifier models where only a handful of simple features are used as the input were also possible. However, they were generally found to be outperformed by the Below Upper Bound baseline, and thus were excluded from the results.

Ground Truth Threshold Set	Popularity Type	Classifier Type	Precision	Recall	F1
$q_U = 0.8$ $q_L = 0.65$ (166 positives in 2019)	Binpop	Two-step	(0.565, 0.648, 0.715)	(0.5, 0.58, 0.681)	(0.555, 0.605, 0.681)
		One-step	**(0.531, 0.603, 0.656)**	**(0.658, 0.72, 0.801)**	**(0.6, 0.658, 0.713)**
	Logpop	Two-step	**(0.486, 0.534, 0.58)**	**(0.739, 0.79, 0.842)**	**(0.587, 0.637, 0.683)**
		One-step	(0.577, 0.631, 0.701)	(0.516, 0.59, 0.72)	(0.546, 0.616, 0.693)
	Rawpop	Two-step	(0.549, 0.616, 0.736)	(0.499, 0.55, 0.64)	(0.533, 0.594, 0.66)
		One-step	(0.444, 0.493, 0.588)	(0.619, 0.69, 0.761)	(0.528, 0.573, 0.655)
	–	Baseline Previous year	(0.375, 0.434, 0.481)	(0.399, 0.44, 0.481)	(0.392, 0.436, 0.481)
	–	Baseline Below Upper Bound	(0.404, 0.448, 0.495)	(0.8, 0.86, 0.921)	(0.537, 0.591, 0.639)
$q_U = 0.8$ $q_L = 0.7$ (116 positives in 2019)	Binpop	Two-step	(0.489, 0.619, 0.68)	(0.371, 0.543, 0.714)	(0.468, 0.579, 0.677)
		One-step	(0.446, 0.5, 0.607)	(0.51, 0.643, 0.689)	(0.485, 0.554, 0.643)
	Logpop	Two-step	**(0.4, 0.463, 0.502)**	**(0.684, 0.771, 0.859)**	**(0.511, 0.592, 0.633)**
		One-step	**(0.489, 0.559, 0.609)**	**(0.513, 0.586, 0.686)**	**(0.507, 0.570, 0.644)**
	Rawpop	Two-step	(0.457, 0.523, 0.636)	(0.314, 0.486, 0.6)	(0.399, 0.506, 0.578)
		One-step	(0.405, 0.462, 0.556)	(0.429, 0.514, 0.629)	(0.434, 0.493, 0.578)
	–	Baseline Previous year	(0.225, 0.288, 0.346)	(0.227, 0.329, 0.373)	(0.23, 0.307, 0.346)
	–	Baseline Below Upper Bound	(0.377, 0.427, 0.474)	(0.599, 0.714, 0.773)	(0.48, 0.533, 0.579)
$q_U = 0.7$ $q_L = 0.65$ (99 positives in 2019)	Binpop	Two-step	**(0.472, 0.54, 0.741)**	**(0.2, 0.367, 0.502)**	**(0.307, 0.431, 0.572)**
		One-step	(0.056, 0.2, 0.45)	(0.032, 0.067, 0.167)	(0.04, 0.101, 0.229)
	Logpop	Two-step	**(0.36, 0.424, 0.503)**	**(0.565, 0.667, 0.738)**	**(0.449, 0.519, 0.598)**
		One-step	(0.165, 0.317, 0.573)	(0.065, 0.1, 0.168)	(0.087, 0.165, 0.24)
	Rawpop	Two-step	(0.393, 0.517, 0.701)	(0.167, 0.3, 0.402)	(0.263, 0.367, 0.474)
		One-step	(0.091, 0.258, 0.503)	(0.033, 0.1, 0.167)	(0.049, 0.145, 0.234)
	–	Baseline Previous year	(0.166, 0.215, 0.301)	(0.197, 0.25, 0.4)	(0.188, 0.235, 0.339)
	–	Baseline Below Upper Bound	(0.225, 0.256, 0.296)	(0.598, 0.667, 0.768)	(0.328, 0.373, 0.427)

Table 5.2 – Confidence intervals for the test-set performance of the trained classifier models versus the baselines for three sets of ground truth thresholds. Each parenthesis is in the format (5th percentile, median, 95th percentile). The classifiers (including the Below Upper Bound baseline) are trained on ground truth from 2018 and tested on ground truth from 2019. For each threshold set, those of the trained models that significantly beat all baselines in terms of F1 (based on a Kruskal-Wallis test, significance level of 0.05) are in bold font. The thresholds q_U and q_L are the percentiles (between 0 and 1) used for getting the upper and lower bounds, respectively. For example, 0.8 means the 80th percentile.

	Reference classifier (243 predicted positives)	Below Upper Bound baseline (310 predicted positives)
🟩 Kubernetes	+	+
🟩 Kotlin	+	−
🟩 AR/VR	+	+
🟩 Tensorflow	+	+
🟩 Keras	−	+
🟥 Logstash	−	+
🟥 Apache Cordova	+	−
🟥 Bigquery	−	−
🟥 Numpy	−	+
🟥 D3.js	−	+
🟥 Cryptocurrency	+	+

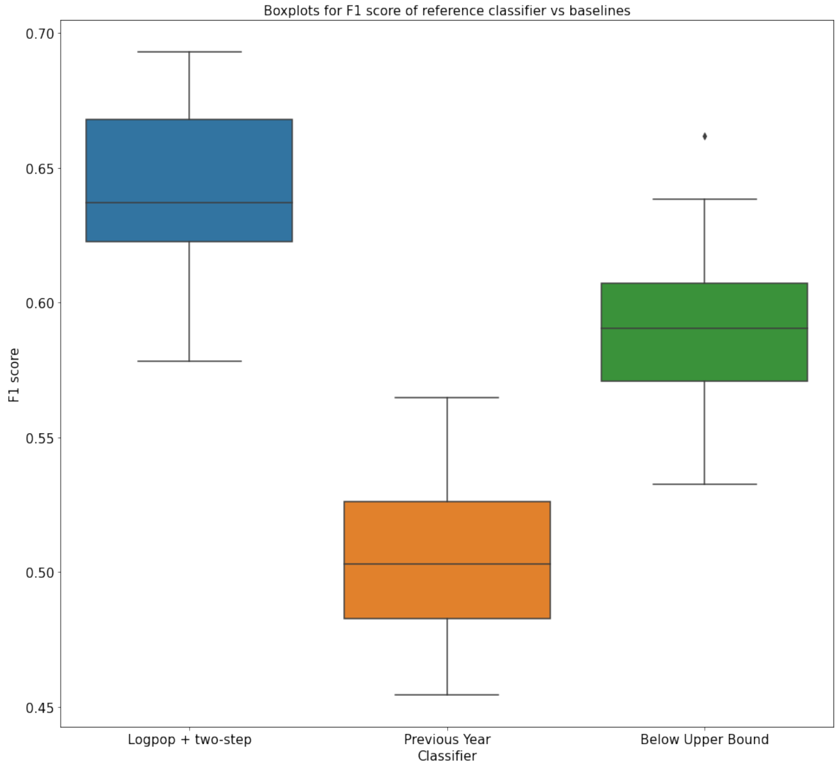
Table 5.3 – Examples of skills predicted as positive (+) or negative (−) by the reference classifier (Logpop+two-step; $q_U = 0.8$, $q_L = 0.65$) versus the corresponding Below Upper Bound baseline. The skills have been selected to be recognizable and are not randomly sampled. The skills in green are ground truth positives; i.e., emerging (per the definition of emerging skills, and for the thresholds $q_U = 0.8$ and $q_L = 0.65$), while those in red are ground truth negatives.

reference classifier achieves better performance than the baseline while predicting over 20% fewer skills as emerging, and is therefore much more suitable for the goal of providing experts with insights.

When it comes to a comparison of the different model types, the best-performing model across the board is Logpop + two-step, while Binpop + two-step also generally shows good performance. One for one (i.e. keeping every other factor constant), Rawpop classifiers fails to outperform any Logpop or Binpop classifiers. This has a very interesting implication: hiring spread is a very important component in predicting hiring volume. Comparing one-step and two-step classifiers shows that the former fail badly for the threshold set $(0.7, 0.65)$, showing the greater robustness of the two-step classifier for different threshold sets.

Predicting the further future

It is now time to move on to the second research question, which is the question of whether classification performance drops when trying to predict further into the future. It is a rather difficult question to answer with the data available in this study, since its length is limited to 3 years. To answer it, let us define **first-half emerging skills** as follows: skills that are emerging when only considering hiring demand in the first half of the year and deleting the second

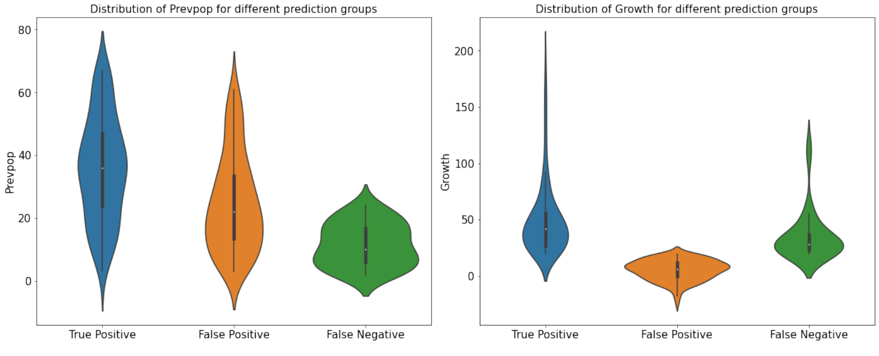


(a)

Figure 5.2 – Boxplots comparing the F1 score of the 20 Logpop + two-step classifiers versus the respective Previous Year and Below Upper Bound baselines for $q_U = 0.8$ and $q_L = 0.65$.

half of the year from the calculations. In a similar way, let us define **second-half emerging skills**. Now, first-half-only emerging skills are defined as those that are first-half emerging but not second-half emerging, and second-half-only skills are defined as the inverse[4]. The first-half-only emerging skills of 2019 should, intuitively, be easier for the classifier models to predict than the second-half-only emerging skills of 2019, as the latter are essentially being predicted 6 months further into the future. Unsurprisingly, by reviewing the predictions of the reference classifier, this intuition is found to be true. The true positives correctly predict 38 out of 47 of the first-half-only emerging skills, while only predicting 23 out of the 39 second-half-only emerging skills. A chi-squared test to see if the recall on second-half emerging skills is significantly different from recall on the rest rejects the null hypothesis (i.e. the recall being the same) at a significance level of 0.01, whereas the same test for the first-half emerging skills fails to reject the null hypothesis. Therefore, the conclusion is that **performance does deteriorate significantly when trying to predict further into the future**,

[4] Bear in mind that, although unlikely, it is possible for first-half-only or second-half-only emerging skills to *not* be emerging when considering the whole year. In the present analysis, only the emerging ones are considered.



(a) (b)

Figure 5.3 – The distributions of **(a)** $Prevpop$ and **(b)** $Growth$ values for true positives, false positives, and false negatives of the reference classifier (on the test set, meaning that these are the rawpop values of these skills in 2018). Note that $Prevpop$ values are non-negative.

making this an important direction for future improvement. However, when it comes to the utility of the presented results for course material creation in the ICT sector, this does not present much of an issue. This is due to the fact that as the previous chapter showed, course material creation in ICT takes, in the median case, less than 1 month, which is the smallest unit of time in the present study. Therefore, if the prediction of ICT emerging skills is *at all successful*, it will be useful to training providers and educators regardless of performance deterioration further into the future.

5.4.2 The limits of predictive performance

The third research question concerns the relationship between the specific definition of emerging skills (or in other words, the values of q_U and q_L) and the performance of the classifier models. As the results in Table 5.2 showed, reducing q_U led to a considerable worsening of performance across the board. The fact that the threshold set $(0.7, 0.65)$ differs from the threshold set $(0.8 and 0.65)$ only in terms of q_U, which sets the upper bound on $Prevpop$, suggests that the models are generally worse at predicting the less popular emerging skills.

To see whether or not this is true, the true positives, false positives, and false negatives of the reference classifier are investigated by examining their $Prevpop$ and $Growth$ values. Figure 5.3 shows the violin plots of these distributions. According to Figure 5.3a, the $Prevpop$ values of the reference classifier's false negatives (with a median of 10) are generally much lower than the $Prevpop$ values of the true positives (with a median of 22) and false positives (with a median of 36), and the $Prevpop$ distribution of false negatives is much more different from the $Prevpop$ distribution of true positives, compared to the $Prevpop$ distribution of false positives. Meanwhile, Figure 5.3b shows that when it comes to $Growth$ values, the false negatives (with a median of 28) are more similar in distribution to the true positives (with a median of 46), whereas false positives have much lower values (with a median of 6). For both $Prevpop$ and $Growth$, the distribution for the false negatives is significantly different from that of the true positives (Kruskal-Wallis test, significance level of 0.01), but the effect size is much smaller in the case of $Growth$ distributions. All this evidence means that the set of false negatives is comprised of skills whose annual growth was relatively comparable to the true positives, but whose previous-year popularity was much lower. This resulted in much weaker signals from their past job ad time series, which caused the reference classifier to incorrectly classify them as negatives. In other words, their surge in popularity emerged *too rapidly* for the model to appropriately detect. Therefore, the answer to the third research question is that **classification performance deteriorates for skills whose past popularity is too low**, and this is another area for future improvement, which shall be discussed further later.

5.4.3 Important features

The last research question is concerned with feature importance. For this analysis, the coefficients of the reference classifier model are investigated (made possible by the fact that all the classifiers used in this study are linear models). These features are themselves linear combinations of the original time series features (since part of the feature reduction pipeline is PCA). To compute an ad-hoc importance score for each original feature, its coefficient in each of the model's features is multiplied by the weight of that feature in the model, and these values are summed up. Then, the original features are ranked using this ad-hoc score. The ranked features and their scores can be seen in Appendix E (Table E.1). Based on the values seen in this table, the two most important families of features that contribute positively to the "emergence" of a skill are as follows:

- Features pertaining to non-linearity, sudden growth, and spikes, such as **the number of data points below the mean**, **the value of the time reversal asymmetry statistic**, **skewness**, and **the longest strikes below and above the mean**.
- Features related to the amount of variation in the time series, such as **variance**, **mean absolute sum of changes**, and **whether the variance is larger 1**.

The features that contribute negatively to emergence are murkier in general. The most important family is **the number of recurring data points**, which would penalize time series where many of the values are the same number. Some nonlinearity features show up as negative contributors, but the positive contributors of that family outweigh the negatives.

Looking at these positive and negative contributors, it seems that the most important signals of skill emergence are sudden growth, spikes, and generally larger variation, which is intuitively expected, given the definition of emerging skills. This is also consistent with the false negative problem that was previously discussed: when a skill's job ad counts are low, almost all of the positive features will have reduced values, making it much more likely for the skill to be classified as non-emerging.

5.5 Discussion

5.5.1 Implications for educational institutions

The results of this study showcase the feasibility of forecasting emerging skills in the ICT domain, although work remains to be done on its generalizability to other ICT labor markets and other professional domains, which will be dealt with in the next chapter. This success shows that AI can help enable educational institutions to keep up with rapid changes in the labor market, especially since the ICT domain is among the most rapidly evolving professional domains. Although the methodology developed in this study does not have perfect precision or recall, it often only fails to predict the emergence of much less popular skills, which are usually (but not always) related to larger skills that it does classify as emerging (e.g. even though Keras is not predicted as emerging, TensorFlow is). This, plus the fact that this method predicts a manageable number of skills as emerging means that it is able to provide insightful information about the evolution of the skills landscape to training providers and decision makers. In short, this is a human-in-the-loop methodology whose results are meant for expert eyes, not direct use. For example, the early warning provided by this method allows training providers to create short, skill-based online courses in anticipation of the emergence of particular skills, thus speeding up curricular change. Here, the unique focus of the presented methodology on less popular and more granular skills (which are the ones more likely not to have already been identified as important skills) provides an additional advantage: finer-grained skills require shorter courses, which would in turn be quicker to make.

5.5.2 Limitations

The presented pipeline and its results have a few limitations. Firstly, the available data was only three years long, meaning that ground truth emerging skills could be computed only for two years (2018 and 2019). As a result, the only test of future prediction was to train models that predict one year into the future for one specific year. The fact that the prediction of the emerging skills of 2019 achieved good performance means that after the preprocessing steps, the skill time series from 2017–2018 and those from 2018–2019 look reasonably similar. In other words, the changes in the market from the year 2017 to the year 2018 are not so big as to make the signals learned based on 2017 data useless for 2018 data, and the classifier model is able to pick up signals that are relevant for both years. However, it is not known whether this phenomenon would occur for other periods, since the forecasting ability of this methodology has only been tested for a single pair of years. Additionally, the model's predictive ability deteriorates when predicting further into the future. However, if training programs can be created rapidly enough, this will not be a big issue. Such a time frame is not unreasonable for skill-based MOOCs due to their smaller size than full-fledged training programs, as shown in Study 2.

The second important limitation of the present work is that it ignores all trends that are larger in scope than individual skill trends. This means that it ignores the relationships between skills and is oblivious to larger trends, such as the collective rise of a group of skills (e.g. the simultaneous rise of several deep learning-related skills) or the rise of one group accompanied by the fall of another (e.g. a new wave of JavaScript-based technologies phasing out older web technologies).

Finally, as discussed before, this methodology struggles to correctly predict emerging skills whose previous popularity is too low. In other words, it does not work well when a skill emerges very quickly and unpredictably (although this term may be a bit of a tautology here). This is a limitation of job ad data, and its impact on the value of the results depends on the experts that wish to use them: if they only consider lower-popularity skills to be truly emerging, then the impact of this issue will be larger. Therefore, addressing this problem is a high priority for future work on emerging skills methodology.

5.5.3 Future directions

This study opens up multiple avenues for future work, both in the form of generalizations and improvements to the existing system.

First is the generalization of this methodology to other domains. Since the methodology is self-contained, with the emerging skill ground truth and the predictive signals all coming from job ads, it is, *on paper*, generalizable to any professional domain and any labour market where formal job ads exist. The main question, when it comes to generalizability, is whether emerging skills are a viable practical concept in the professional domain in question. The most important factor in answering this question is the rate at which the labour market evolves, both in terms of the appearance of new skills and the growth of existing skills. This is something that was essentially taken for granted in the ICT domain, since it had already been proven to be rapidly-evolving in Study 2. The proposed methodology provides a framework for verifying the viability of emerging skill prediction through the question of whether or not baseline emerging skill predictors are beaten, and a study of the professional domain's rate of evolution would further strengthen this methodology's ability to verify the applicability of the concept of emerging skills to said domain. This will be explored further in the next chapter.

Secondly, since the results imply the importance of hiring spread in the prediction of hiring volume, a follow-up question is as follows: Is there a particular set of companies that anticipate skill trends well? In other words, is a skill's spread among certain companies more important than its spread among others? The idea that such a set of companies exists makes intuitive sense in the ICT domain, where Big Tech are often the creators of new technologies, and following these corporations alone can yield valuable insights into the direction of the market in the near future. This idea could provide an improvement to the proposed pipeline: An approach where the spread of a skill among companies is weighted by the "influence" of each company (as opposed to the current approach, where every company has the same weight), with more "trend-anticipating" companies having larger weights. The "influence" concept would have to be defined based on the company's past ability to predict emerging skills.

There are several research directions directed towards rectifying the limitations of the current pipeline. One of the limitations of the proposed models is that they predict the emergence or non-emergence of each skill only using its own job ad trends, essentially ignoring the relationships between skills. In reality, many skills are related, and their trends are part of larger trends. For example, the simultaneous rise of "Tensorflow", "Keras", and "PyTorch" is not accidental, but rather due to the rise of "Deep Learning" in general. This points towards an approach incorporating a skills ontology: if the relationship between the four skills mentioned above is made explicit (e.g. through "Deep Learning" being a parent of the other three), it could be incorporated into a model that looks not only at the job ad trends of the skill it's predicting, but also at those of its parents and children. Such a classifier model would need to be more complex than the linear models used in this study.

Another limitation to address is the fact that the current pipeline only uses one source of data. On one hand, this is a strength, since it makes it self-contained and applicable to any domain. On the flip side, however, auxiliary data sources that are domain-specific could provide additional signals and improve our predictive ability. For the ICT domain, an interesting auxiliary data source is Stack Overflow, especially since previous work has shown that it tends to be faster than job ads at manifesting new skills. Incorporating signals from such auxiliary sources could improve the models' ability to forecast skill trends. In the specific case of Stack Overflow, however, preliminary tests of a model using both job ad data and Stack Overflow data were unsuccessful and attained lower performance than models only using job ad data, likely due to the considerable differences between the Singaporean ICT market and the Stack Overflow space, which is global and does not reflect skill demand, but rather skill *learning* demand in a complicated fashion (due to the closure of repeated questions).

5.6 Conclusion

This study presented a generalizable methodology to predict emerging skills, showing its feasibility in the ICT domain, which is one of the fastest-changing domains. This methodology's early identification of rising skills provides training providers and domain experts with insights that help speed up the process of curricular change, thereby allowing educational institutions to keep up with the trends and to equip workers with the right skills for a changing labor market. Therefore, AI is a double-edged sword, disrupting labor markets but also able to help institutions and people adjust to the new markets, thus addressing some of the problems it causes. Future work incorporating skill ontologies and auxiliary signals can help address the limitations of the presented method and push the boundaries of emerging skill prediction even further, providing more accurate and more comprehensive insights to experts.

5.7 Acknowledgements

This research would not have been possible without the availability of fine-grained job ad data. Gratitude is due both to JobTech, who are the owners of the data and agreed to its sharing, and to SkillsFuture Singapore, who made it available for this study. In addition, due thanks to SkillsFuture Singapore for their indispensable input and feedback during the research process.

6 Study 4: Emerging Skills in VET

6.1 Introduction

In the previous chapter, the idea of “emerging skills” was found to have practical merit, the prediction of emerging skills was shown to be feasible, and a combination of hiring spread and hiring volume were found to perform best when predicting near-future emerging skills. The important question that emerged out of this successful approach was the questions of how to capitalize on this success. Three directions existed for further work. The first was the improvement of the emerging skills prediction pipeline for the ICT sector, using the data utilized in the previous chapter. The second direction was to expand on the idea of hiring spread and “trend-anticipating” firms. The idea would be to investigate whether or not a detection of such firms would allow for the accurate prediction of emerging skills, using the firms they were adopted by first as the signal. Finally, the third approach was to try to generalize the emerging skills approach to a few vocational education domains and gauge whether the success of the previous study could be replicated in a different professional domain with different dynamics. After some deliberation, the latter was found to be the direction with the greatest potential impact, and was chosen for this chapter. Therefore, the goal of this chapter is to gauge the feasibility of emerging skills prediction in a few domains that fall under the vocational education and training (VET) umbrella, and to characterize the differences between these domains and the ICT domain.

6.2 Related work

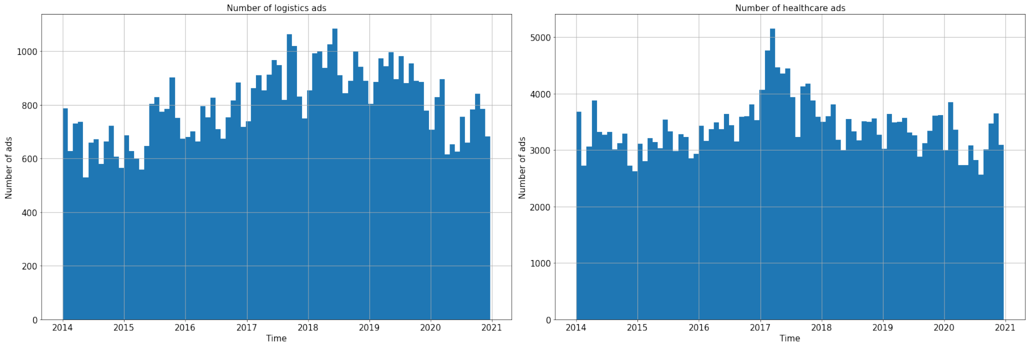
6.2.1 VET and Switzerland

Vocational Education and Training refers to a very diverse educational field whose main goals are to prepare individuals for their working life, and to provide them with training throughout their working life, potentially including lifelong learning and support when switching occupations (Billett, 2011; Markowitsch and Hefler, 2019). The systems used for VET in different countries are much more diverse than those used for other educational fields, but a common arrangement used in Germany, Switzerland, and Austria (among a few others) is a dual system where in-school education is combined with on-the-job training in the form of apprenticeships (SKBF and CSRE, 2018). In the Swiss VET system, students spend only 1-2 days at school, and spend the rest of the week at their apprenticeship. This system, which requires close collaboration between the teachers at school and the managers at the workplace, covers a wide range of professional domains, from traditional trades (such as carpentry, plumbing, electrical and mechanical work, etc.) to healthcare, information technology, and clerical jobs. Therefore, this system covers blue-collar and white-collar work, and traditional and new jobs alike. It is held in high international regard due to its demonstrated success and its important contribution to the Swiss economy. Its high reputation among the Swiss public stems from the fact that it gives students working responsibilities from an early age, provides hands-on learning, gives them a well-recognized degree that can get them jobs, and leaves their options open with the avenues it has for switching to more academic pathways (Hoffman and Schwartz, 2015).

As mentioned in the first chapter, the latest Swiss education report on the Swiss VET system (SKBF and CSRE, 2018) states that per current laws, each professional domain has one committee dedicated to the development and quality assurance of its VET program, including the curriculum. The members of these committees are the partners of the VET system (i.e. from firms and schools that participate in the VET system as partners) and they are in charge of making sure the curriculum is up to date by reviewing (and potentially revising) the curriculum at least once every five years. According to the report, an average of around 20 changes have been made to all the Ordonnances de Formation Professionnelles (professional training ordinances) combined, per year (SKBF and CSRE, 2018). These changes are often small changes to the regulations or curricula, [1].

The rapid pace of change brought about by Industry 4.0 and other disruptors begs the following question: is once every five years enough? As Studies 2 and 3 showed, 5 years is woefully slow for the ICT domain, where even one year can bring considerable changes to the skills landscape. However, although some ICT sub-fields fall under the VET umbrella (e.g. “mediamatician”), other VET domains may be slower-evolving and may not require methodologies that deal with rapid skill change. This is why it is important to investigate whether the concept of emerging skills is practical in VET domains and to study the timescales involved in VET. At the same time, the large number of VET domains and the differences between them mean that it is important to focus on only a few of them in order to keep the study itself feasible.

[1] The ordinances and their change logs can be found on https://www.becc.admin.ch/becc/public/bvz/beruf/grundbildungen



(a)

(b)

Figure 6.1 – The number of job ads per month in the JobCloud dataset in (a) logistics and (b) healthcare, 2014-2021.

6.3 Objectives and Methodology

6.3.1 Study Objectives

The objectives of this study are as follows:

- To try out the emerging skills methodology in a number of VET domains and to evaluate its performance, comparing it to the results of the previous chapter
- To quantify the conditions that make such a classification feasible and to analyze what these conditions say about the domain in question

To this end, two VET domains have been chosen for this study: logistics, and healthcare. IT skills are relatively prevalent in these two domains, which means that they are expected to feel the impact of Industry 4.0 more strongly and be subjected to more rapid changes. In addition, both domains have relatively large numbers of job ads, and this can be crucial in the success of a predictive methodology. This is why these two domains were chosen, rather than slower-evolving and/or smaller domains such as gardening or carpentry.

6.3.2 Data

The data in this study comes from the Swiss job portal JobCloud, who have kindly shared their data for the purpose of this research project. The data includes every single job ad posted on each of JobCloud's job boards: jobup.ch, and jobs.ch. The former is mostly for companies in French-speaking Switzerland, whereas the latter is mostly for those in the German-speaking part; both cover a wide variety of domains, from ICT to manufacturing to many different VET

Domain	Logistics	Healthcare	ICT
Time period	2014-2020	2014-2020	2017-2020
Total # of ads	516,219	1,346,598	563,200
Avg. # of ads / year	86,037	224,433	187,733
# of skills with >= 1 ad / year *	2,068	2,500	864
Median # of ads per skill (for >= 1 ad / year skills)	22	29	48
Avg. # of new skills per year (2017-2020) **	258	281	159
# of companies	4,845	4,595	2,264

Table 6.1 – Summary statistics for the two datasets in this study, compared with those of the ICT dataset from the previous study. * The skills considered in logistics and healthcare are all hard skills, whereas those in ICT are only IT skills manually labelled out of the full set of hard skills. ** A new skill in a year y is defined as a skill that has at least one ad in year *y*, but no ads in year *y-1*. Only skills with at least one ad per year are considered here.

domains. The overwhelming majority of these ads are in German, French, or English, with a small minority being in Italian and other languages. The data available for both domains is from 2014 to 2021. Figure 6.1 shows the monthly number of job ads in the two domains for the entire period. In order to exclude the effects of COVID-19 (which is expected to have made 2020 very different from the previous years), only the data from 2014-2020 is included in this study.

Preprocessing

Before the data can be used in the emerging skills methodology, two steps are necessary: the translation of every job ad to English, and the extraction of skills from those ads. For the translation step, the Python package langdetect is used to detect the language of each job ad, and then MarianMT models (part of HuggingFace transformers) (Junczys-Dowmunt et al., 2018) are used to translate them into English. The extraction of skills is trickier, since it needs very high recall (i.e. the method should not *miss* the skills in an ad) and has to deal with imperfections in the previous translation step. Several methods were tried on a small sample of 40 job ads: the Python package Skills-ML, Microsoft Azure's Named Entity Recognition API, and the EMSI Skills API. The results of this trial strongly favored the EMSI Skills API, which had very high precision *and* recall (whereas the other two methods had very low recall), and was therefore chosen as the skill extraction method. After discussions with EMSI, they kindly provided free unlimited access to their API, allowing for its full use as the skill extraction part of the preprocessing pipeline.

Summary statistics on the resulting datasets for logistics and healthcare can be seen in Table 6.1, side by side with summary statistics on the ICT dataset used in the previous chapter. As these values show, there is no shortage of new skills in the two domains under study, although the ratio of new skills to all skills is lower than in ICT. In addition, the median demand (i.e. number of job ads) per skill is around half that of ICT in Logistics and Healthcare, despite the fact that ICT is being considered in the 2017-2020 period of 3 years, whereas the other two are being considered in the 2014-2020 period of 6 years. This means that more low-demand skills are to be expected in these two domains. However, based on the number of new skills per year, there is no reason to believe that these domains do not evolve rapidly enough for the emerging skills methodology to be useful, although there is reason to believe that it would have lower performance given the prevalence of lower-popularity skills.

6.3.3 Methodology

The methodology used in this study is essentially the same as that of the previous study: one year-long time series are used to predict whether a skill is emerging or non-emerging in the subsequent year, and thresholds on $Prevpop$ (q_U) and $Growth$ (q_L) are used to define emerging skills. Since the two-step, linear regression-based classifier consistently outperformed the one-step classifier in the previous study, this study will solely use the former, although the different popularity types (Rawpop, Logpop, and Binpop) will all be tested.

The previous study showed two limitations in the methodology:

- The predictive methodology has a hard time correctly predicting emerging skills with low $Prevpop$, since the time series of those skills contain weaker predictive signals.
- Emerging skills with lower $Growth$ values are more likely to become false negatives.

As a result of these limitations, and in order to account for the large number of low-demand skills in the two datasets of this study, two heuristics are used to make the conditions of the predictive task for each dataset similar to that of the previous study:

- Skills are sorted by $Prevpop$, and those with their $Prevpop$ below a threshold $Prevpop_{min}$. are removed from the training and test sets, essentially truncating the set of skills. This ensures that the training and test sets are not saturated with noisy, low-demand skills. The value of $Prevpop_{min}$ has been set to 5 in this study[2].
- The values of q_U and q_L (the quantile thresholds) are set such that the numerical values of U and L (the numerical thresholds) are similar (but not necessarily equal) to the

[2] Lower and higher values were tested as well to see how sensitive the results were to this value, and their sensitivity was found to be relatively low. The general takeaways in the results section hold as long as the value of this lower bound is no greater than 10, with the reference classifier's performance improving once this threshold is raised above 10 (which eliminates a very large number of skills)

	Dataset (q_U, q_L)	Logistics (0.95,0.95)				Healthcare (0.9, 0.88)				ICT (0.8, 0.65)			
Training year - Test year	Classifier type	Prec.	Rec.	F1	# pos.	Prec.	Rec.	F1	# pos.	Prec.	Rec.	F1	# pos.
2018-2019	Ref. clf.	0.389	0.082	0.136	18	0.171	0.271	0.210	152	0.581	0.772	0.663	210
	Baseline (BUB)	0.158	0.412	0.229	220	0.226	0.521	0.315	210	0.460	0.905	0.610	310
	Baseline (PY)	0.065	0.077	0.070	64	0.080	0.072	0.076	75	0.433	0.446	0.439	171
2017-2018	Ref. clf.	0.300	0.090	0.138	20	0.247	0.169	0.201	85	-	-	-	-
	Baseline (BUB)	0.223	0.403	0.287	120	0.168	0.677	0.269	501	-	-	-	-
	Baseline (PY)	0.096	0.098	0.097	52	0.048	0.060	0.053	83	-	-	-	-
2016-2017	Ref. clf.	0.375	0.111	0.171	16	0.267	0.095	0.140	30	-	-	-	-
	Baseline (BUB)	0.199	0.519	0.287	140	0.192	0.345	0.247	150	-	-	-	-
	Baseline (PY)	0.118	0.100	0.108	51	0.149	0.123	0.135	67	-	-	-	-

Table 6.2 – The classification results (precision, recall, F1, and number of predicted positives) for logistics and healthcare for three different periods: 2016-2017, 2017-2018, and 2018-2019. The results for ICT in 2018-2019 are included for comparison. In all cases, the lower and upper bounds discussed in the Methodology section are used. The reference classifier is the best-performing classifier trained using the proposed method: for logistics and healthcare, this is a **Binpop**-based classifier (since it outperformed Rawpop and Logpop), whereas for ICT, it is a **Logpop**-based classifier. "Baseline (BUB)" is the **Below Upper Bound** baseline, while "Baseline (PY)" is the **Previous-Year** baseline, both of which were extensively discussed in the previous chapter, with the former being the best performing baseline developed in this methodology.

previous study's U and L for $q_U = 0.8$ and $q_L = 0.65$ across all years. These values have been chosen because the best performance of the method in ICT was attained with them. For Logistics, the chosen quantile thresholds are $q_U = 0.95$ and $q_L = 0.95$, whereas in Healthcare, they are $q_U = 0.9$ and $q_L = 0.88$.

6.4 Results

6.4.1 Emerging skill classification

The results of the reference classifiers for healthcare and logistics can be seen for three different years, side by side with the results for ICT in 2018-2019, in Table 6.2. The results (some of which have been omitted from the table for brevity, and are simply discussed below) are striking in multiple ways:

- The performance measures are very low across the board. While F1 scores in ICT are around 0.6, F1 scores in logistics and healthcare are often around 0.25. This is because both precision and recall are greatly affected, and the baseline seems to suffer more in terms of precision while the reference classifier seems to suffer more in terms of recall.
- In logistics and healthcare, the reference classifier *always* loses to the Below Upper

Bound baseline.

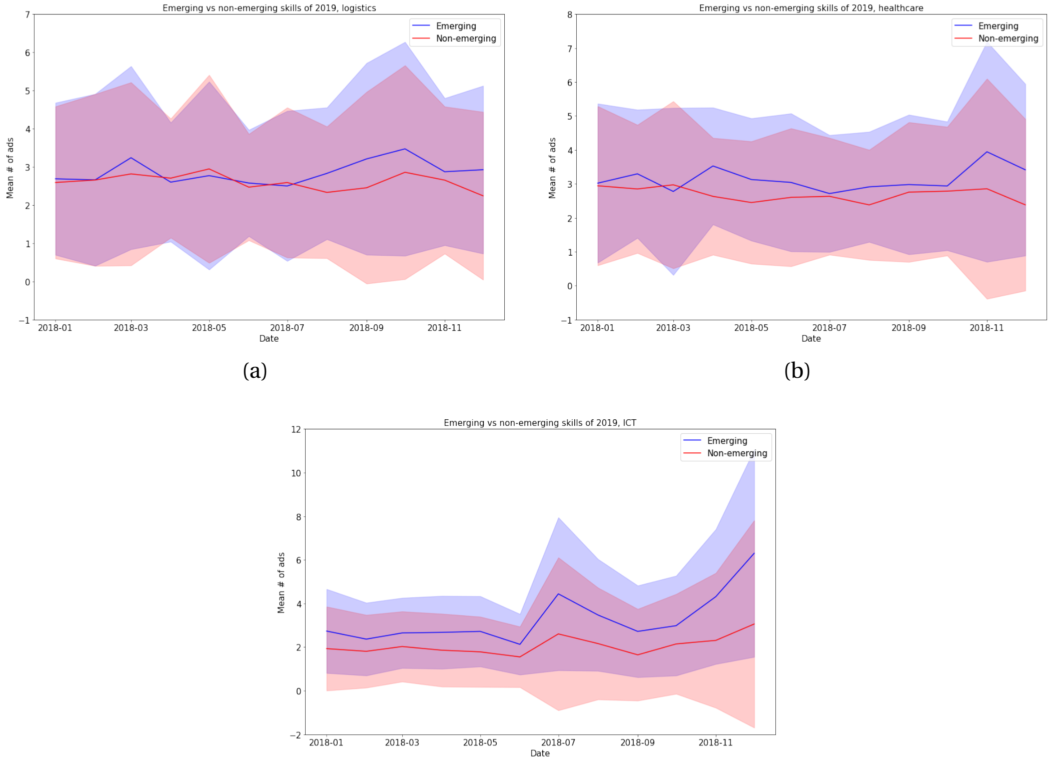
- In most cases, in both logistics and healthcare, the reference classifier predicts many fewer skills as emerging than the baseline, which corresponds to the fact that there are far fewer ground truth emerging skills in logistics and healthcare than there are in ICT when using the same numerical thresholds Q and L.
- In every case, the Previous-year baseline (which predicts the emerging skills of the previous year as the emerging skills of this year) is beaten by the reference classifier. This is because the intersection of the emerging skill sets of subsequent years is very small (often around 6-7), particularly compared to ICT where close to half of the emerging skills of 2018 were also emerging in 2019.
- A result that has been omitted from the table for the sake of brevity is that Binpop-based classifiers are the best performing ones in logistics and healthcare, rather than Logpop-based classifiers. In addition, the best performance is obtained by using the original Binpop time series, rather than the features extracted from them (which lead to even more overfitting).

Given the observed results, the rest of the results section will deal with two questions: why the performance of the classifier is so much worse in logistics and healthcare, and how its performance could be improved.

6.4.2 Investigating the reduced performance

One thing that is evident in the results in Table 6.2 is that even the *baseline* – which is always the best-performing classifier in logistics and healthcare – performs much worse in logistics and healthcare than it does in ICT. The baseline's F1 score in those two domains is always around 0.3, whereas in ICT, it is 0.610. To interpret this difference, let us discuss what the baseline really does.

The Below Upper Bound baseline sorts the skills by their $Prevpop$ value, removes any that are above q_U, and then takes the top N skills and predicts them as emerging. Therefore, its underlying assumption is that **the more popular a skill has been in the previous year, the more likely it is to emerge in the next**. As a result, the meaning of the observed difference in performance is that this assumption is *much less true* in logistics and healthcare than it is in ICT. Another piece of evidence that points towards this explanation is the correlation between $Prevpop$ and $Growth$ in the three domains, shown in Table 6.3. As the values show, the relationship between emergence and $Prevpop$ is very tenuous in both logistics and healthcare (and when considering all skills, even the *sign* of the relationship can change between different years!), whereas there is a clear and strong relationship between the two in ICT. What this



(a)

(b)

(c)

Figure 6.2 – The average time series for emerging versus non-emerging skills for **(a)** Logistics, **(b)** Healthcare, and **(c)** ICT, for the year 2018 (i.e. the emerging vs non-emerging skills of 2019). The line indicates the average value and the colored area indicates average ± standard deviation.

implies about how hiring happens in logistics and healthcare is that it is a lot less "steady" and lot "burstier": instead of a skill gradually increasing in demand until it experiences a surge and attains "emerging" status, a skill may experience a surge with very low previous demand, making the surge more "abrupt", informally speaking. In addition, the smaller number of emerging skills in these two domains, especially after using the $Prevpop_{min}$ threshold to filter out the ones with very low previous demand, results in an underperforming classifier since its number of positive samples is too low for it to learn enough about signals that differentiate between emerging and non-emerging skills. Figure 6.2 paints a similar picture that supports this interpretation. In ICT, emerging skills exhibit clear growth in demand over time compared to non-emerging skills. However, in the other two domains, such growth is either non-existent or very limited, and the only things that separate the two groups of skills from one another are that the average demand for emerging skills is slightly higher and slightly spikier than that of non-emerging skills.

	Logistics		Healthcare		ICT	
Year	2018	2019	2018	2019	2018	2019
Pearson Correlation, $Prevpop$ and $Growth$ (All skills)	0.64	-0.64	-0.90	0.02	0.73	0.95
Spearman Correlation, $Prevpop$ and Emerging label (Skills above $Prevpop_{min}$ only)	0.16	0.11	0.05	0.16	0.43	0.53

Table 6.3 – Correlations between $Prevpop$ and $Growth$ (first row) and emerging label (0 or 1, second row) for the years 2018 and 2019 in the three domains.

	ROC-AUC value					
	Logistics		Healthcare		ICT	
Training year - Test year	Ref. clf.	Baseline	Ref. clf.	Baseline	Ref. clf.	Baseline
2018-2019	**0.614**	**0.611**	0.462	0.751	**0.797**	**0.790**
2017-2018	0.660	0.680	**0.622**	**0.613**	-	-
2016-2017	0.688	0.715	0.644	0.713	-	-

Table 6.4 – The ROC-AUC values for the reference classifiers and baselines for all three domains in the 2016-2017, 2017-2018, and 2018-2019 periods.

Another interesting observation (which is excluded from Table 6.2 for the sake of brevity) is that Binpop performs better than Logpop in these two domains. This may imply that hiring volume is even less useful here than it was in ICT for predicting emerging skills. However, since the reference classifier is always beaten by the baseline in logistics and healthcare, this is not much of a practically useful insight. In order to see whether this is really the case, an alternative baseline classifier was created that would use binarized $Prevpop$ instead of raw $Prevpop$ in order to predict emerging skills, and the resulting baseline classifier was found to perform much worse, with the original baseline attaining a training set F1 of 0.24, and the modified one attaining an F1 of 0.10. Therefore, rather surprisingly, the best overall classifier for both Logistics and Healthcare seems to be a *baseline* that uses *Rawpop.*

All of these obervations lead to an important question: is it at all possible to create conditions (e.g. constraints), under which the reference classifier is able to beat the baseline in the two domains of this study, and would it still be useful under those conditions? The first of these questions will be answered in the next section, while the second will be answered in the Discussion section afterwards.

Training year - Test year (Domain)	Classifier Type	Prec.	Rec.	F1	# pos.
2018-2019 Logistics	Ref. Clf.	0.193	0.329	0.243	145
	Baseline	0.158	0.412	0.229	220
2017-2018 Healthcare	Ref. Clf.	0.189	0.484	0.271	315
	Baseline	0.168	0.677	0.269	501

Table 6.5 – The performance of the two reference classifiers that beat the respective baselines after adjusting q_{pred}.

6.4.3 Improving classifier performance

In the previous section, the results showed that the reference classifier was consistently beaten by the baseline. One path for further investigation is the fact that the two-step classifier uses a threshold (which shall be called q_{pred}, whose default value is q_L) to declare the skills with their predicted growth above that threshold as emerging and those below as non-emerging. However, the reference classifier seems to predict much fewer skills as emerging compared to the baseline when $q_{pred} = q_L$. Therefore, decreasing q_{pred} may be one way to help improve the classifier's performance. In order to see how much this would help, the ROC-AUC values for all the reference classifiers and baselines have been computed, which can be seen in Table 6.4. The results show that some of the reference classifiers that are beaten by the respective baseline using the default configuration have a better ROC-AUC value, meaning that they are, across different threshold values, better at distinguishing between emerging and non-emerging skills. Therefore, a change in the q_{pred} threshold may, at times, enable the reference classifier to outperform the baseline. In addition, the gap between the ROC-AUC value of the reference and baseline classifiers is often much smaller in Logistics than in Healthcare, meaning that the reference classifier tends to perform much worse for Healthcare.

By using q_{pred} values that result in the reference classifier predicting around two-thirds as many positives as the baseline, the results in Table 6.5 are achieved, in which the reference classifier beats the baseline, although just barely. Therefore, the proposed classifier mostly fails to beat the Below Upper Bound baseline, and performance (both that of the baseline and the proposed classifier) is quite low.

6.5 Discussion

6.5.1 Why the classifier failed

Having presented all the results of the predictive approach, it is important to summarize the reasons why this prediction was ultimately unsuccessful:

1. Despite the number of "new" skills in logistics and healthcare being comparable to that of ICT, the relationship between previous demand and demand growth is much weaker and often *inverted* (i.e. less previous demand corresponds to greater growth!). This means that the fundamental assumption underlying the emerging skills approach – which is that emerging skills grow gradually up to some point (corresponding to the early adoption period in the innovation diffusion S-curve) and then experience a surge in demand – is violated in these two domains. In other words, emerging skills emerge "abruptly" instead of "gradually", which weakens the signals available in their past job ad time series and makes the prediction task very difficult. The fact that there are very few overlaps between the emerging skills of subsequent years further reinforces this idea: most of the skills that "emerge" in year y but do not go above the q_U threshold do not continue to grow enough to be considered emerging in year $y+1$ as well, further demonstrating the bursty and abrupt growth of skill demand in logistics and healthcare.

2. The number of emerging skills in healthcare and logistics is much lower, often around 70-80 rather than the 150-160 in ICT, which means that even given similar numbers of overall data points in the training and test sets, there are much fewer positive data points for the classifier to learn from. This leads to overfitting even when using no more than 5 to 10 features for the classifier, which was not the case in ICT.

An analysis of why these two problems exist (i.e. why the relationship between $Prevpop$ and $Growth$ is so weak, and why there are fewer emerging skills) is beyond the scope of this chapter (in particular due to time constraints). However, an important difference between the two domains in this study and ICT is that the Industry 4.0 innovations that drastically impact ICT labor markets and change the skills landscape *originate from ICT itself*, whereas they are exogenous to logistics and healthcare. This can help explain why there are fewer emerging skills in these domains, and why the behavior of emerging skills is bursty: the innovations do not originate from within the domain, which means that it is less likely to see a pattern where the creators of the innovation adopt it first, followed by its spread among other firms.

6.5.2 Emerging skills still matter

The most important question raised by the unsuccessful prediction of emerging skills in logistics and healthcare is whether or not emerging skills are still a valuable concept in these domains. If they cannot be reasonably predicted, then what is their use? Here, it is important to go back to the definition of an emerging skill, which is a skill that used to have low demand, and has suddenly experienced a surge in demand. This definition means that, regardless of predictability, emerging skills are an important concept, as they are the skills most likely to have been previously ignored by training providers. Even if a domain has only 10 emerging skills altogether, they still matter. The pertinent question is whether or not a machine learning methodology is required (or useful) for their prediction and for helping the expert decision makers. Here, the answer is more ambiguous. This study has shown that a complex predictive methodology such as the one proposed here may fail to accurately predict emerging skills in some domains, especially if the number of emerging skills is low in the domain in question and thus hampers machine learning algorithms. In such a case, methodologies based on expert opinions or focus groups may be preferable, although certain insights can still be learned using the methodology proposed here. In particular, and ironically, the Below Upper Bound baseline seems to be the best-performing method (or rather, heuristic) for predicting emerging skills in the two domains of this study, and it can be used instead of the proposed classification pipeline in order to provide experts with insights. Although this is not an ideal solution, it can still be a useful one if the baseline predicts a manageable number of emerging skills that can be effectively screened by experts. This usefulness is only reinforced by the fact that the baseline often has a (relatively) low false negative rate and a high false positive rate, and when expert screening is involved, keeping the former low is much more important than keeping the latter low.

6.5.3 Future work

Although the proposed approach resulted in an underperforming classifier, there are several ways in which its performance may be improved, which were not tried due to time constraints. All of these directions for future work mentioned in the previous chapter apply here as well, often more strongly. Here, three of them will be discussed in particular. The first potential remedy is the incorporation of a skills hierarchy. An unsupervised version of this was attempted for the present study, but the clustering failed to improve classification performance. However, a hierarchy created (either entirely or partially) by experts may still help improve performance. The second way to improve performance would be to hand-pick the features of the classifier. As previously discussed, the best performing features in this study were the original Binpop time series, rather than the extracted features, which indicates that the feature selection pipeline did not work as intended. Nevertheless, even the original Binpop time series failed to produce a sufficiently accurate classifier. Therefore, manually choosing the features to be used in the classifier is another approach that could be tried in order to improve performance. Finally, the third way to improve performance could be to incorporate other data sources into the classifier. As mentioned before, one reason why the classifier underperformed in logistics and healthcare is that most of the disruptive skills (which have appeared thanks to Industry 4.0) are exogenous. Incorporating data from the domain those skills originally come from can be a remedy to the problem arising from the bursty nature of hiring demand for emerging skills in these domains.

6.6 Conclusion

This study explored the application of the emerging skills methodology developed in the previous chapter to logistics and healthcare, two of the domains that fall under vocational education. The results showed that the methodology was unsuited to these domains due to the dynamics of these domains, which resulted in fewer emerging skills and spiky behavior rather than steady but non-linear growth. Despite these setbacks, the concept of emerging skills remains useful, and further work is required on the application of the classification methodology to domains other than ICT in order to either tailor it to the particularities of these domains, or to rule out its feasibility in them.

7 Discussion

The previous chapters have discussed the research approaches taken in this dissertation, their results, and their implication for the subsequent studies. This chapter is dedicated to summarizing the contributions of the studies conducted as part of the present work, discussing their implications for educators and training creators, and synthesizing them into methodologies and actionable insights that can be used by educators in various contexts.

7.1 Summary of results

The contributions (both positive and negative) of the present dissertation to the analysis and identification of training needs are as follows:

1. An approach using Stack Overflow response times to determine the relative difficulty of learning different topics and using a survey of practitioners to verify the computed relative difficulties was unsuccessful, showcasing, firsthand, the level of subjectivity involved in surveying practitioners about the skills of their domain, and the importance of breaking the difficulty of a topic down to more fine-grained categories.

2. The software programming domain was shown to be a very fast-evolving domain, where tens of granular new skills appear every year. In this domain, among the online platforms for education and hiring, those that are more grassroots and less institutional were shown to be quicker at manifesting the new skills that appear in the software programming domain (which pertains to the first appearance of the skill on each platform). Both the Q&A platform Stack Overflow and the MOOC platform Udemy were shown to manifest these new skills faster than job postings on Stack Overflow Jobs do, the former by a median of around half a year, and the latter by a median of around a month. However, the signals found on Stack Overflow were insufficient for the prediction of the

first appearance of new skills on Udemy or Stack Overflow Jobs using Stack overflow. This lack of success shaped the development of the emerging skills approach.

3. The concept of “emerging skills”, i.e. previously less demanded skills that have recently had an uptick in hiring demand, is shown to be a useful practical concept for analyzing the labor market and the skill needs of its workforce in the ICT domain. The prediction of the emerging skills of the near future using the past job posting trends of those skills was shown to be feasible, as it managed to beat all the alternative baseline methods. This allows for the ahead-of-time provision of valuable insights on the near future demand of less popular skills to educators, who can use this information to create training materials for those skills in advance. A mixture of hiring volume (i.e. the raw number of job postings for a skill) and hiring spread (i.e. the number of firms hiring for a skill) is found to perform best for predicting emerging skills.

4. Emerging skills are shown not to be a panacea for skill need identification, as the proposed classification methodology fails to provide good accuracy in predicting the emerging skills of the Logistics and Healthcare domains. The main culprit is revealed to be the fact that previous demand and demand growth are much less correlated in these two domains compared to the ICT domain, providing a feasibility checking heuristic for establishing whether or not the emerging skills approach is likely to succeed in a particular domain. This situation is possibly due to the fact that many of the emerging skills in these two domains come from *outside* of the domain (since they are IT skills), whereas in ICT, they used to come from *within* the domain itself.

7.2 Contributions

7.2.1 Emerging skills: the present and the future

Emerging skills and training material creation

Study 3 presented the emerging skills prediction methodology, which was successful in the context of the Singaporean ICT domain. Study 4 then applied this approach to VET domains, resulting in the addition of feasibility checks to the beginning of the pipeline. In order to make this methodology fully usable for the purpose of training material creation by educators, it is necessary to consolidate it within a complete framework that not only establishes feasibility checks for the prediction task, but also allows educators to evaluate whether the predictions are accurate enough to be useful in practice, and to then put them to use. To create this framework, the emerging skills methodology is combined with parts of the descriptive approach from Study 2. Figure 7.1 shows the proposed framework for feasibility checking emerging skills prediction, performing said prediction, and putting those predictive insights into practice.

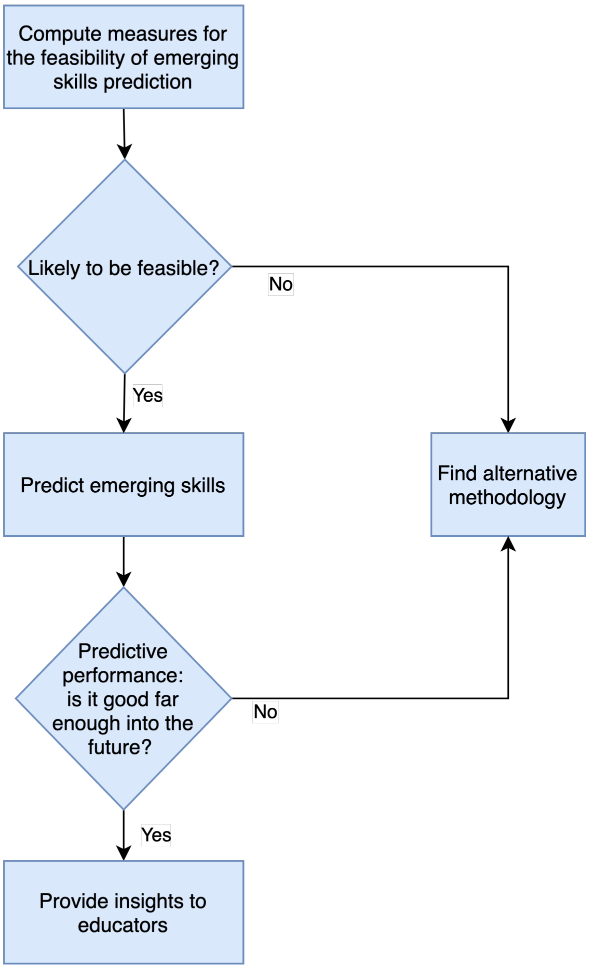

Compute measures for the feasibility of emerging skills prediction
Likely to be feasible?
No
Yes
Predict emerging skills
Find alternative methodology
Predictive performance: is it good far enough into the future?
No
Yes
Provide insights to educators

Figure 7.1 – The proposed framework for how educators can use the emerging skill methodology.

The first step is the computation of several heuristics that help establish whether or not the prediction of emerging skills is going to be successful in the selected domain. These measures consist of the following, in decreasing order of importance:

1. The correlation between previous demand and demand growth for all skills, which helps determine whether or not emerging skills *look different enough* from non-emerging skills. If the correlation is low or negative, it means that the fundamental underlying assumption of the approach – that emerging skills have a phase of gradual growth before their popularity explodes – is violated, and the prediction is unlikely to succeed.
2. The overall number of skills that exist in that domain and the average number of new skills entering the domain annually, since the classifier needs enough positive examples

to learn properly, and the set of all skills is always a relatively small set.

If the prediction is deemed to have basic feasibility, the second step is to actually perform the prediction, in a fashion identical to the methodology used in Study 3 and Study 4. Here, the evaluation step is what determines whether or not the results of the prediction are useful, and is the ultimate feasibility check. First of all, the emerging skills classifier needs to be able to beat the baselines, since the baselines represent simple ways of predicting emerging skills, and if the trained classifier is unable to beat them, the reasons need to be diagnosed. If the classifier does beat the baselines, then the important question is **"How far into the future is the prediction superior to that of the baselines"?** The reason why this question matters is because the goal of this prediction is essentially to give educators advance insights into the important skills of the near-future, with enough of a head-start so that they would have time to create training materials for those skills ahead of time. This way, once the skill actually emerges, the training materials will already exist, allowing workers to make use of it immediately as the skill experiences a surge in hiring demand. Therefore, the main measure of interest here is **the time it takes to create training materials for a skill**. Of course, this value is different for each skill, since some skills require less training materials and/or are generally easier to learn. Therefore, what matters is some summary statistic of this quantity (e.g. the median or some percentile above 50), the choice of which depends on how much of a head start is desired by the educators. If the prediction is accurate further into the future than this value, then it gives the educator ample time to make a decision about creating training materials for each skill (or group of skills), and to create them if necessary. In case of Study 3, an estimate of the amount of time it takes to create an ICT MOOC (on Udemy) was computed in Study 2 (although what was computed in Study 2 was for *new* skills, not *emerging* skills). Therefore, it is possible to gauge whether or not the predictions are useful in practice. As Study 2 showed, the median time for the creation of ICT courses on Udemy is about 10 days. Due to the possibility of such fast course creation, essentially *any* successful emerging skill prediction task in the ICT domain would be effective in providing advance insights for over half the skills.

The proposed framework is most suitable for fast-changing domains with a large number of skills. If at any step of this process, it is determined that the prediction of emerging skills is not feasible in the professional domain or labor market in question, or if predictive performance is not good enough far enough into the future, then a different methodology needs to be chosen for that domain or market, perhaps from among the existing and more traditional methodologies for skill need identification. Due to the limited time available for Study 4, much remains to be explored in terms of evaluating this methodology in other professional domains and labor markets, and that work is left to future researchers.

Other uses of emerging skills

The utility of emerging skills for educators is not limited to the use of their prediction for the ahead-of-time creation of training materials. It is certainly possible that by the time a skill emerges, training materials already exist for it (either on decentralized platforms such as Udemy, or even by more centralized institutions). In such a case, as discussed before, the creation of further training materials is certainly not out of the question, since the existing courses may not satisfy the requirements of the educators in question. In addition, knowledge of emerging skills gives educators the ability to create up-to-date training materials, which is useful if they prefer to create their own course material rather than refer their students to pre-existing ones.

Going further, the prediction of emerging skills does not even necessarily have to lead to training material creation. If suitable previous training material exists for the skills that are expected to emerge in the near future, the insights gained from the prediction of emerging skills can be used to advertise and promote education on those skills. For such a use case, the emerging skills do not necessarily need to be predicted as far into the future, since the head start needed for course creation is not necessary here.

Finally, as discussed in the previous chapter, even if the *prediction* of emerging skills is unsuccessful in a certain professional domain, the concept remains useful as an analytic tool. This is evidenced by the analyses of emerging skills (under other names) in existing literature (Strack et al., 2020).

7.2.2 Decentralized platforms for education

To practitioners in the software programming domain, the popularity of decentralized platforms such as Stack Overflow may come off as natural: Stack Overflow has, for almost a decade, been one of the main learning hubs for software programmers (Anderson et al., 2012). However, the present work has shown that the value of Stack Overflow goes beyond its value for the individual developer, as its decentralized quality results in extraordinary agility in manifesting new topics in the software programming domain. As Study 2 showed, a majority of new skills appear on both Stack Overflow and Udemy before they do on Stack Overflow Jobs, with median delays being around 8 months for Stack Overflow and 1.5 months for Udemy. In other words, these new skills appear on decentralized educatinal platforms before they appear in job ads, even though the latter are the main indicator of labor market demand for a skill.

In the present dissertation, this agility was used in order to analyze the dynamics of hiring and education and to understand skill trends, and later to help support the emerging skills approach. However, in the greater scheme of things, there is a case to be made for *supporting and promoting* decentralized educational platforms themselves - both Q&A platforms and decentralized MOOCs - in other professional domains, especially in domains where digital technologies have had less penetration.

The existence of a Q&A platform in a professional domain that connects novices and experts with each other is a win-win for both practitioners and learners on one side, and educators on the other: the former gain an instrument of lifelong learning that allows them to expand their knowledge, while the latter gain insights into the skills landscape of the domain without needing to collect any data, simply from the online traces of practitioners' learning behaviors. Such insights would allow them to aggregate the skill needs found on the website, and create training programs based on them if needed. Of course, such an outcome would first require the Q&A platform to become popular, and that is not a given, as some professional domains may not be as well-suited as others to such a platform due to a variety of reasons (e.g. how comfortable the practitioners are with digital technologies, how tangible the questions are, etc.). In addition, unless the platform enjoys a certain degree of popularity, the skill needs found on the platform cannot be taken as a proper representation of the actual skill needs of practitioners. In any case, a look at the list of current Stack Exchange websites[1] reveals that although technology-related Q&A forums are by far the most popular (with Stack Overflow having had 21 million questions), there are also Q&A forums for gardening (16k questions), interpersonal skills (3.8k questions), and aviation (21k questions), among others. Therefore, it is not impossible for Q&A forums to become popular for such topics.

Decentralized MOOC platforms are helpful for learners in a similar way. They allow practitioners with expertise in a particular skill to share that expertise with other practitioners by directly teaching them that skill. This model involving the establishment of a direct link between practitioners with varying degrees of expertise is broadly similar to how Q&A forums operate, and its popularity can be seen in the large number of diverse courses on Udemy, covering subjects from software programming to gardening. This means that essentially, **decentralized platforms for informal learning can help fill the gaps created by changing skills, making the prediction of important future skills (which is primarily important for formal learning platforms) less important**.

The main difference in case of such MOOC platforms is in how they can be used to give insights to educators. These platforms give training providers an understanding of the skills landscape of the domain by looking at the supply of courses by expert practitioners and the demand for those courses by learners, aggregating these into insights on the whole domain. At the same time however, the creation of a MOOC on a particular skill means that its importance has been recognized by *someone*, which can be a person (in Udemy's case) or an institution, and this could potentially reduce the urgency of creating training materials for that skill. This is closely related to the fact that as Study 2 showed, content creation on decentralized MOOC

[1] https://stackexchange.com/sites

platforms is more expert driven than content creation on Q&A platforms, meaning that the signals appear on decentralized MOOC platforms at a later point in the S-curve of diffusion. However, the existence of a previous course for a skill on a MOOC platform like Udemy does not mean that educators do not need to create training materials for it. There may be a need for more up-to-date course materials, the course may need to adhere to a standardized format, or there may even be competition between said educators and the MOOC platform where the previous course has been published. Therefore, the insights that MOOC platforms can provide to educators are as useful as those provided by Q&A forums.

7.3 Future work

The future work that can be done based on this dissertation falls into two categories: policy-related work on the promotion of decentralized educational platforms, and further work on the emerging skills approach. As the former has been discussed in detail in the previous section, this section will focus on the latter. Especially since the main contribution of this dissertation is the emerging skills approach, it is worthwhile to reiterate and re-contextualize the directions in which this methodology can be expanded, which were first discussed in Chapter 5 and then expanded upon in chapter 6.

The first direction is the use of auxiliary data sources as additional input signals. For example, Stack Overflow could be used as an additional source of input signals in the ICT domain (although as Study 3 showed, it may not be a good match for regional job ad data), and so could other online big data sources. For domains other than ICT, the choice of auxiliary sources is less obvious, but Q&A and MOOC platforms can nevertheless be of use if they are popular enough. This is a relatively easy modification to implement, as the auxiliary signals can simply be fed to the classifier as inputs just like the job ad data.

When using auxiliary data sources for additional signals, it is quite important to consider the different contexts of these datasets, as they are almost never from the same labor market as the job ad data being used: Stack Overflow and Udemy are both global in context, which could potentially mean that they have very different skill trends. In addition, the internal dynamics of these platforms are quite different from job ads. For example, in job ads, what matters the most is the demand that various companies have for a skill, whereas on a platform like Stack Overflow, the discouragement of duplicate questions means that just because a topic is not getting many new questions does not mean that it is in low demand among learners: it could simply be that many of the relevant questions already exist. [2] In addition, conceptually, *demand by employers* and *demand by learners* are two different things, and they do not necessarily correspond to the same skill trends.

[2] This problem essentially boils down to a lack of granular information about view counts on Stack Overflow, which was discussed in previous chapters.

The second direction is the use of auxiliary data sources as structural additions. An important example of this would be integrating a skills ontology into the classifier, so that it would associate the trends of related skills with each other and recognize higher-level skill trends. Another way to expand the model structurally would be to devise a way to integrate economic census data into the classifier. Such an approach could make the model aware of higher-level economic trends, allow it to better account for noisy fluctuations in hiring, and help it see higher-level trends similar to how a skills ontology would. Finally, as Study 4 showed, it may be much more useful to study a professional domain alongside the domains that are the source of the new and emerging skills entering the former. This would necessitate a model that can simultaneously predict emerging skills for different domains while transferring knowledge from one domain to the other. Obviously, all of these proposed additions would require a fundamental redesign of the classification pipeline for emerging skills, as the current design simplifies the problem by assuming that the different skills are independent and identically distributed variables (which they are not).

The last direction is weighting each company with an "influence" score based on how trend-setting or trend-anticipating it is. Since Study 3 showed that a combination of hiring volume and hiring spread works best for predicting emerging skills, the natural next step is to make the hiring spread "company-aware". Such an approach would mean that a skill being demanded by a more trend-anticipating company (i.e. an early adopter or innovator) counts as a stronger signal of potential emergence, compared to said skill being demanded by a less trend-anticipating (i.e. a late adopter) company, which makes intuitive sense. This approach has the potential downside of disadvantaging *new* trend-anticipating firms, which are trend-anticipating even though the data does not yet show it.

7.4 Conclusions

The present dissertation has presented a series of methodologies for analyzing and identifying training needs on the professional scale, synthesized into a single framework for investigating the dynamics and evolution of a labor market followed by the prediction of emerging skills, which are the potential training needs of the near future. The insights provided by this framework can be used by educators and training providers, not only for the creation of training material in advance, but also for promoting the right skills for the workforce ahead of time. This allows for the rapid evolution of training curricula and help educational institutions keep up with the pace of change brought about by Industry 4.0 as well as other disruptors. In addition, the evidence provided in this dissertation for the agility of decentralized educational platforms calls for their larger scale adoption and promotion, especially since they can provide educators with large-scale insights directly coming from the learners themselves.

A Features used in the difficulty estimation classifier, Study 1

This appendix shows the features used in Study 1's classifier.

Feature of	Feature name	Polynomial degree
User posting the question	Total # of questions	1
	Sum of upvotes on questions	1
	Average upvote on questions	2
	Mean upvote on questions	1
	Sum of downvotes on questions	1
	Mean downvote on questions	2
	Median downvote on questions	1
	# of questions with accepted answer	1
	# of questions with any answer	1
	Total # of answers on questions	1
	Accepted answer median time	2
	Accepted answer mean time	2
	Accepted answer maximum time	2
	Accepted answer minimum time	1
	First answer median time	2
	First answer mean time	2
	First answer maximum time	2
	First answer minimum time	1
User answering the question (accepted)	Total # of questions	2
	# of accepted answers	3
	# of all answers	3
	Accepted answer median time	4
	Accepted answer mean time	3
	Accepted answer maximum time	4

Feature of	Feature name	Polynomial degree
	Accepted answer minimum time	1
	Sum of upvotes on answers	3
	Mean upvote on answers	2
	Median upvote on answers	2
	Sum of downvotes on answers	3
	Mean downvote on answers	2
	Median downvote on answers	1
Question	Length of body	2
	Length of title	1
	Title starts with a capital letter?	1
	Monthly impact (based on tags)	2
	Monthly popularity (based on tags)	2
	Length of code	1
	Code to text ratio	1
	Popularity of title (TF-IDF)	2
	Number of tags	2
	Number of URLs	2
	Posted on a weekend?	1
	Posted from midnight to 8 AM?	1
	Posted from 8 AM to 4 PM?	1
	Posted from 4 PM to midnight?	1
Answer	Length of body	2
	Length of code	1
	Code to text ratio	2
	Number of URLs	2

Table A.1 – The features of the the question poster, the responder, the question itself, and the answer itself used in Study 1's classifier. These features, plus the topics, make up all the features of the classifier. The polynomial degree used for each feature is indicated (the topics all have a degree of 1).

B Matching Stack Overflow tags to their occurrences in text, Study 2

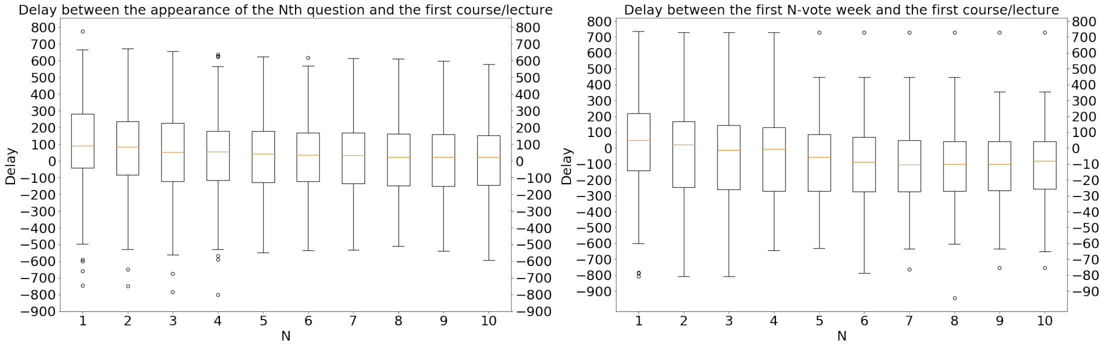
When matching of tags to their occurrences in Udemy lecture and course titles and Stack Overflow Jobs ad descriptions in Study 2, it was necessary to pay attention to several details in order to increase recall and reduce the number of tags missed (which could considerably alter the results of the study, since the tag numbers are not very high to begin with). First of all, Stack Overflow tags are not necessarily unigrams, and therefore, in order to match text to tags, n-grams need to be generated. To do so, a Porter stemmer was used on both the text and to each unigram in every tag. This made sure that false negatives due to grammatical features (such as a word being plural in the text and singular in the tag) would be avoided. Also, extra care was taken to properly match version tags, i.e. tags like “vuejs2” which refer to a specific version of a technology. Simply, a version tag is matched to the base tag (e.g. “vuejs2” becomes “vuejs”), and minor version tags are matched to the major version ((e.g. “python3.6” becomes “python3”. Some tags happen to be is a substring of another, unrelated tag; for example, a piece of text that matches the tag “tail-recursion” (a concept in algorithms) would also match the tag “tail”, which is a command that prints out the last lines of a file. This problem was dealt with in two stages. First, the following two heuristics were used in order to narrow down the set of possible false positives:

1. Comparing the popularity of the two tags. Usually, the tag which is a substring of the other is the more general one, and thus should have more questions associated with it if they are really related.
2. Checking the co-occurrence of the two tags. Related tags should co-occur often.

Neither of these two heuristics are 100% accurate, therefore necessitating a manual check. Therefore, after narrowing down the possible pairs of unrelated tag-subtag pairs, a manual pass was done on the pairs, and a list was compiled of those that were actually unrelated so that they could then be filtered out at a later time. When such a tag and subtag occured in a text, only the longer tag was kept and and the subtag was discarded, since it is always the subtag that is a wrong match.

C Signals of tag importance, Study 2

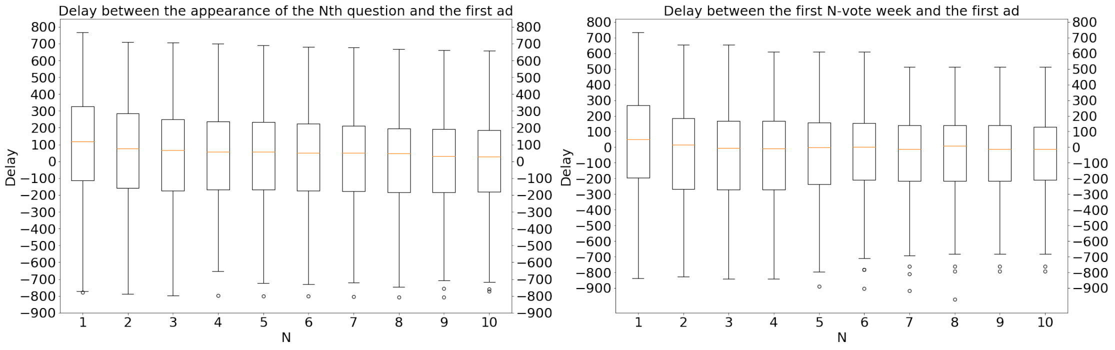
This appendix includes Figures C.1 and C.2, which are boxplots that show how much time there is between the emergence of stronger signals of a tag's importance on Stack Overflow, and its appearance on Udemy or Stack Overflow Jobs.



(a)

(b)

Figure C.1 – Boxplots of the delay (in days) between (a) the appearance of the N-th Stack Overflow question or (b) the first N-vote week of a tag and its first mention on Udemy (with the horizontal axis being N). The horizontal orange line in each box is the median. Each box is only for tags that do have the event (e.g. a 6th Stack Overflow question), hence the occasional *increase* in the median, although the increases are never large. Notice the multiple negative median values for (b), and the decrease from around 100 to 24 for (a).



(a)

(b)

Figure C.2 – Boxplots of the delay (in days) between (a) the appearance of the N-th Stack Overflow question or (b) the first N-vote week of a tag and its first mention on Stack Overflow Jobs (with the horizontal axis being N). Again, the orange horizontal line inside each box is the median for that box. The medians in (a) go from 119 days for N=1 to 27 days for N=10, and medians for (b) again have many negative values.

D Tag themes and types in Study 2

This appendix provides definitions for tag themes and types in Table D.1. The number of tags that has each theme/type is also included. Only themes and types with at least 5 tags have been listed, for the sake of brevity.

Name	Theme or Type?	# of tags	Description
web	Theme	123	Web development
cloud	Theme	62	Cloud technologies
ml	Theme	40	Machine Learning
db	Theme	31	Databases
gp	Theme	24	General-purpose application development
mobile	Theme	24	Mobile development
build	Theme	20	Building/compiling applications, can come alongside various other themes
container	Theme	20	Containers, such as Docker
blockchain	Theme	20	Blockchain and cryptocurrencies
server	Theme	14	Servers and serverless technologies
viz	Theme	12	Tools for the design of animations and images
game	Theme	10	Video game-related topics
microserv	Theme	10	Microservices
vr	Theme	8	Virtual Reality
management	Theme	8	Tools related to managing customers or employees
bot	Theme	7	Chatbots
vision	Theme	6	Computer Vision
network	Theme	5	Network

Name	Theme or Type?	# of tags	Description
test	Theme	5	Testing and test automation tools
ide	Theme	5	Integrated Development Environments
analysis	Theme	5	Tools for analyzing data
solution	Type	141	Full-fledged solutions, such as entire cloud database technologies, Customer Relatonship Management suites, etc.
framework	Type	95	Application frameworks written in a particular language, covering a wide range of applications.
library	Type	72	Libraries within a particular language.
tool	Type	48	These are tools that fulfill a specific purpose but not generally as a programming language or framework, and are small in scope. Tools such as command line interfaces and network proxies fall into this category.
feature	Type	36	Features of a programming language or framework.
connector	Type	32	Technologies that connect others to each other, such as APIs.
concept	Type	20	Concepts. The other types are focused on technologies, but this focuses on concepts not related to one particular technology family.
lang	Type	5	Programming languages (general purpose or specific scripting).

Table D.1 – Tag themes and types that appear at least 5 times in our set of 227 tags, along with the number of tags they are related to and a short description of what they stand for.

E Importance scores for emerging skill prediction features, Study 3

This appendix shows the ad-hoc importance scores computer for each of the original features in the classifier in Study 3.

Original Feature	**Ad-hoc Importance**
cid_ce__normalize_True	3.260370174
last_location_of_minimum	3.041300917
longest_strike_above_mean	1.819125048
time_reversal_asymmetry_statistic__lag_2	1.7711566
skewness	1.746095463
kurtosis	1.721839525
longest_strike_below_mean	1.704475687
minimum	1.490589562
variance_larger_than_standard_deviation	1.470279171
has_duplicate_min	1.470279171
benford_correlation	1.470279171
mean_abs_change	1.453978262
standard_deviation	1.300465484
variance	1.254512743
mean_second_derivative_central	1.218262811
first_location_of_maximum	1.210340323
last_location_of_maximum	1.174841021
median	1.107915982
variation_coefficient	1.050612425
maximum	1.000516271
absolute_sum_of_changes	0.985774444
has_duplicate_max	0.944529448

Original Feature	**Ad-hoc Importance**
sum_of_reoccurring_data_points	0.944529448
abs_energy	0.944529448
c3__lag_3	0.7474519
ratio_value_number_to_time_series_length	0.559784466
c3__lag_1	0.449628367
sum_of_reoccurring_values	0.39138383
first_location_of_minimum	0.258098861
count_above_mean	0.204591848
time_reversal_asymmetry_statistic__lag_3	0.030081356
mean_change	-0.218127186
sum_values	-0.233810196
percentage_of_reoccurring_values_to_all_values	-0.353926206
count_below_mean	-0.52953518
percentage_of_reoccurring_datapoints_to_all_datapoints	-0.571704456
has_duplicate	-0.655769556
mean	-0.850531591
time_reversal_asymmetry_statistic__lag_1	-1.468221572
c3__lag_2	-1.631146116

Table E.1 – A list of features used in the reference classifier, along with their ad-hoc importance scores. Explanations of the features and their names can be found at https://tsfresh.readthedocs.io/en/latest/text/list_of_features.html .

Bibliography

Agrawal, S., Smet, A. D., Lacroix, S., and Reich, A. (2020). To emerge stronger from the COVID-19 crisis, companies should start reskilling their workforces now [White paper]. Technical report, McKinsey Insights.

Allamanis, M. and Sutton, C. (2013). Why, when, and what: Analyzing Stack Overflow questions by topic, type, and code. In *2013 10th Working Conference on Mining Software Repositories (MSR)*, pages 53–56.

Anderson, A., Huttenlocher, D., Kleinberg, J., and Leskovec, J. (2012). Discovering value from community activity on focused question answering sites: a case study of stack overflow. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 850–858. Association for Computing Machinery.

Bai, J. J., Brynjolfsson, E., Jin, W., Steffen, S., and Wan, C. (2020). Digital Resilience: How Work-From-Home Feasibility Affects Firm Performance. SSRN Scholarly Paper ID 3616893, Social Science Research Network.

Barua, A., Thomas, S. W., and Hassan, A. E. (2014). What are developers talking about? An analysis of topics and trends in Stack Overflow. *Empirical Software Engineering*, 19(3):619–654.

Baruch, Y. and Holtom, B. C. (2008). Survey response rate levels and trends in organizational research. *Human Relations*, 61(8):1139–1160.

Bernstein, S., Atkinson, A. R., and Martimianakis, M. A. (2013). Diagnosing the Learner in Difficulty. *Pediatrics*, 132(2):210–212.

BGT (2019). Mapping the Genome of Jobs: The Burning Glass Skills Taxonomy [White paper]. Technical report, Burning Glass Technologies.

Bhat, V., Gokhale, A., Jadhav, R., Pudipeddi, J., and Akoglu, L. (2014). Min(e)d your tags: Analysis of Question response time in StackOverflow. In *2014 IEEE/ACM International*

Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014), pages 328–335.

Billett, S. (2011). *Vocational Education: Purposes, Traditions and Prospects*. Springer Science & Business Media.

Blei, D. M. and Lafferty, J. D. (2006). Correlated topic models. In *Advances in neural information processing systems*, volume 18.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research 3*, pages 993–1022.

Boehm, B. (2006). Some future trends and implications for systems and software engineering processes. *Systems Engineering*, 9(1):1–19.

Borg, G., Bratfisch, O., and Dorni'c, S. (1971). On the Problems of Perceived Difficulty. *Scandinavian Journal of Psychology*, 12(1):249–260.

Bozkurt, A., Keskin, N. O., and Waard, I. d. (2016). Research Trends in Massive Open Online Course (MOOC) Theses and Dissertations: Surfing the Tsunami Wave. *Open Praxis*, 8(3):203–221.

Brynjolfsson, E. and McAfee, A. (2011). *Race Against the Machine: How the Digital Revolution is Accelerating Innovation, Driving Productivity, and Irreversibly Transforming Employment and the Economy*. Brynjolfsson and McAfee.

Brynjolfsson, E. and McAfee, A. (2014). *The Second Machine Age: Work, Progress, and Prosperity in a Time of Brilliant Technologies*. W. W. Norton & Company.

Brynjolfsson, E., Mitchell, T., and Rock, D. (2018). What Can Machines Learn, and What Does It Mean for Occupations and the Economy? *AEA Papers and Proceedings*, 108:43–47.

Buhl, M. and Andreasen, L. B. (2018). Learning potentials and educational challenges of massive open online courses (MOOCs) in lifelong learning. *International Review of Education*, 64(2):151–160.

Cambridge Economics (2019). E3ME - Our Global Macro-Econometric Model (version 6).

Cappelli, P. (2012). *Why Good People Can't Get Jobs: The Skills Gap and What Companies Can Do About It*. University of Pennsylvania Press.

Carnegie, T. A. M. and Crane, K. (2019). Responsive curriculum change: going beyond occupation demands. *Communication Design Quarterly*, 6(3):25–31.

Carter, L. R. (2014). Employers Aren't Just Whining – the "Skills Gap" Is Real. *Harvard Business Review*, 25.

CEDEFOP (2018). Skills Forecast [White paper].

Cen, L., Ruta, D., and Ng, J. (2015). Big education: Opportunities for Big Data analytics. In *2015 IEEE International Conference on Digital Signal Processing (DSP)*, pages 502–506.

Chang, J., Boyd-Graber, J., Gerrish, S., Wang, C., and Blei, D. M. (2009). Reading Tea Leaves: How Humans Interpret Topic Models. In *Advances in Neural Information Processing Systems 22 (NIPS 2009)*, pages 288–296.

Chiu, W., Thompson, D., Mak, W., and Lo, K. (1999). Re-thinking training needs analysis: A proposed framework for literature review. *Personnel Review*, 28(1/2):77–90.

Conache, M., Dima, R., and Mutu, A. (2016). A Comparative Analysis of MOOC (Massive Open Online Course) Platforms. *Informatica Economica*, 20(2/2016):4–14.

Correa, D. and Sureka, A. (2014). Chaff from the wheat: characterization and modeling of deleted questions on stack overflow. In *Proceedings of the 23rd international conference on World wide web*, pages 631–642. Association for Computing Machinery.

Coursera (2019). Global Skills Index 2019 [White paper].

Dawson, N., Rizoiu, M.-A., Johnston, B., and Williams, M.-A. (2019). Adaptively selecting occupations to detect skill shortages from online job ads. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 1637–1643.

Dawson, N., Rizoiu, M.-A., Johnston, B., and Williams, M.-A. (2020). Predicting Skill Shortages in Labor Markets: A Machine Learning Approach. *arXiv:2004.01311 [cs, econ, q-fin]*.

Ebben, M. and Murphy, J. S. (2014). Unpacking MOOC scholarly discourse: a review of nascent MOOC scholarship. *Learning, Media and Technology*, 39(3):328–345.

Ehara, Y., Sato, I., Oiwa, H., and Nakagawa, H. (2012). Mining Words in the Minds of Second Language Learners: Learner-Specific Word Difficulty. In *Proceedings of COLING 2012*, pages 799–814. The COLING 2012 Organizing Committee.

Einav, L. and Levin, J. (2014). The Data Revolution and Economic Analysis. *Innovation Policy and the Economy*, 14:1–24.

Ellis, S. P. (2003). Anticipating employers' skills needs: the case for intervention. *International Journal of Manpower*, 24(1):83–96.

Estelles, E., Moral, E. D., and González, F. (2010). Social Bookmarking Tools as Facilitators of Learning and Research Collaborative Processes: The Diigo Case. *Interdisciplinary Journal of E-Learning and Learning Objects*, 6(1):175–191.

Fan, W. and Yan, Z. (2010). Factors affecting response rates of the web survey: A systematic review. *Computers in Human Behavior*, 26(2):132–139.

Field, J. (2000). *Lifelong Learning and the New Educational Order*. Trentham Books, Ltd.

Fischer, G. (2000). Lifelong Learning—More Than Training. *Journal of Interactive Learning Research*, 11(3):265–294.

Forbes (2019). Tech Experts Predict 13 Jobs That Will Be Automated By 2030. https://www.forbes.com/sites/forbestechcouncil/2019/03/01/tech-experts-predict-13-jobs-that-will-be-automated-by-2030/.

Fowler, D., Poling, N., Anthony, W., Morgan, J., and Brumbelow, K. (2014). Data-driven curriculum redesign in civil engineering. In *2014 IEEE Frontiers in Education Conference (FIE) Proceedings*, pages 1–9.

Fritz, T., Begel, A., Müller, S. C., Yigit-Elliott, S., and Züger, M. (2014). Using psycho-physiological measures to assess task difficulty in software development. In *Proceedings of the 36th International Conference on Software Engineering*, pages 402–413. Association for Computing Machinery.

Gallivan, M. J., Truex, D. P., and Kvasny, L. (2004). Changing patterns in IT skill sets 1988-2003: a content analysis of classified advertising. *ACM SIGMIS Database: the DATABASE for Advances in Information Systems*, 35(3):64–87.

Gienapp, L., Stein, B., Hagen, M., and Potthast, M. (2020). Estimating Topic Difficulty Using Normalized Discounted Cumulated Gain. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2033–2036. Association for Computing Machinery.

Goldfarb, A., Taska, B., and Teodoridis, F. (2021). Could machine learning be a general purpose technology? A comparison of emerging technologies using data from online job postings. SSRN Scholarly Paper ID 3468822, Social Science Research Network.

Gopalakrishnan, S. and Damanpour, F. (1997). A review of innovation research in economics, sociology and technology management. *Omega*, 25(1):15–28.

Gould, D., Kelly, D., White, I., and Chidgey, J. (2004). Training needs analysis. A literature review and reappraisal. *International Journal of Nursing Studies*, 41(5):471–486.

Greer, J. and Thompson, C. (2016). Data-Driven Programmatic Change at Universities: What works and how. In *PCLA @ LAK*, pages 32–35.

Gurcan, F. and Cagiltay, N. E. (2019). Big Data Software Engineering: Analysis of Knowledge Domains and Skill Sets Using LDA-Based Topic Modeling. *IEEE Access*, 7:82541–82552.

Hall, S., Stephens, J., Parton, W., Myers, M., Harrison, C., Elmansouri, A., Lowry, A., and Border, S. (2018). Identifying Medical Student Perceptions on the Difficulty of Learning Different Topics of the Undergraduate Anatomy Curriculum. *Medical Science Educator*, 28(3):469–472.

Hammond, M. (2019). A Review of Recent Papers on Online Discussion in Teaching and Learning in Higher Education. *Online Learning*, 9(3):9–23.

Handel, M. J. (2012). Trends in Job Skill Demands in OECD Countries [White paper]. Technical report, OECD.

Hanrahan, B. V., Convertino, G., and Nelson, L. (2012). Modeling problem difficulty and expertise in stackoverflow. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work Companion*, pages 91–94. Association for Computing Machinery.

Haskel, J. and Holt, R. (1999). Anticipating future skill needs: can it be done?: does it need to be done? | VOCEDplus, the international tertiary education and research database.

Hiranrat, C. and Harncharnchai, A. (2018). Using Text Mining to Discover Skills Demanded in Software Development Jobs in Thailand. In *Proceedings of the 2nd International Conference on Education and Multimedia Technology*, pages 112–116. Association for Computing Machinery.

Hoffman, N. and Schwartz, R. (2015). *Gold Standard: The Swiss Vocational Education and Training System. International Comparative Study of Vocational Education Systems.* National Center on Education and the Economy.

Horton, J. J. and Tambe, P. (2015). Labor Economists Get Their Microscope: Big Data and Labor Market Analysis. *Big Data*, 3(3):130–137.

Hosen, A. and Alfina, I. (2016). Aggregation of open data information using linked data: Case study education and job vacancy data in Jakarta. In *2016 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, pages 579–584.

Hughes, J. (2021). krippendorffsalpha: An r package for measuring agreement using krippendorff's alpha coefficient. *arXiv preprint arXiv:2103.12170*.

Illanes, P., Lund, S., Mourshed, M., Rutherford, S., and Tyreman, M. (2018). Retraining and reskilling workers in the age of automation [White paper]. Technical report, McKinsey & Company.

ILO and OECD (2018). Approaches to anticipating skills for the future of work [White paper]. Technical report, International Labour Organization & Organization for Economic Cooperation and Development. 2nd Meeting of the Employment Working Group, Geneva, Switzerland.

Ishola, O. M. and McCalla, G. (2016). Detecting and Supporting the Evolving Knowledge Interests of Lifelong Professionals. In Verbert, K., Sharples, M., and Klobučar, T., editors, *Adaptive and Adaptable Learning*, pages 595–599.

Johnston, S. C. (2018). Anticipating and Training the Physician of the Future: The Importance of Caring in an Age of Artificial Intelligence. *Academic Medicine*, 93(8):1105–1106.

Johri, V. and Bansal, S. (2018). Identifying Trends in Technologies and Programming Languages Using Topic Modeling. In *2018 IEEE 12th International Conference on Semantic Computing (ICSC)*, pages 391–396.

Junczys-Dowmunt, M., Grundkiewicz, R., Dwojak, T., Hoang, H., Heafield, K., Neckermann, T., Seide, F., Germann, U., Fikri Aji, A., Bogoychev, N., Martins, A. F. T., and Birch, A. (2018). Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Kim, Y.-H. (2002). A State of Art Review on the Impact of Technology on Skill Demand in OECD Countries. *Journal of Education and Work*, 15(1):89–109.

Laal, M. and Salamati, P. (2012). Lifelong learning; why do we need it? *Procedia - Social and Behavioral Sciences*, 31:399–403.

Latchem, C. (2016). Learning Technology and Lifelong Informal, Self-directed, and Non-formal Learning. In *The Wiley Handbook of Learning Technology*, pages 180–199. John Wiley & Sons, Ltd.

Lee, K. and Mirchandani, D. (2010). Dynamics of the Importance of IS/IT Skills. *Journal of Computer Information Systems*, 50(4):67–78.

LinkedIn (2019). Future of Skills 2019 Report | LinkedIn Talent Solutions [White paper].

Maisiri, W., Darwish, H., and Dyk, L. (2019). An Investigation of Industry 4.0 Skills Requirements. *South African Journal of Industrial Engineering*, 30:90–105.

Markowitsch, J. and Hefler, G. (2019). Future developments in Vocational Education and Training in Europe: Report on reskilling and upskilling through formal and vocational education training. Working Paper 2019/07, JRC Working Papers Series on Labour, Education and Technology.

Matsuda, N., Ahmed, T., and Nomura, S. (2019). Labor Market Analysis Using Big Data: The Case of a Pakistani Online Job Portal. SSRN Scholarly Paper ID 3491253, Social Science Research Network.

Michaels, G., Natraj, A., and Van Reenen, J. (2014). Has ICT Polarized Skill Demand? Evidence from Eleven Countries over Twenty-Five Years. *The Review of Economics and Statistics*, 96(1):60–77.

Müller, S. C. (2015). Measuring Software Developers' Perceived Difficulty with Biometric Sensors. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, volume 2, pages 887–890.

Newton, G. and McCunn, P. (2015). Student perception of topic difficulty: Lecture capture in higher education. *Australasian Journal of Educational Technology*, 31(3).

Okebukola, P. A. and Jegede, O. J. (1989). Students' Anxiety towards and Perception of Difficulty of some Biological Concepts under the Concept-mapping Heuristic. *Research in Science & Technological Education*, 7(1):85–92.

Papoutsoglou, M., Mittas, N., and Angelis, L. (2017). Mining People Analytics from StackOverflow Job Advertisements. In *2017 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pages 108–115.

Ponzanelli, L., Mocci, A., Bacchelli, A., Lanza, M., and Fullerton, D. (2014). Improving Low Quality Stack Overflow Post Detection. In *2014 IEEE International Conference on Software Maintenance and Evolution*, pages 541–544.

Porter, S. R., Whitcomb, M. E., and Weitzer, W. H. (2004). Multiple surveys of students and survey fatigue. *New Directions for Institutional Research*, 2004(121):63–73.

Prinsley, R. and Baranyai, K. (2015). *STEM Skills in the Workforce: What do Employers Want?* Office of the Chief Scientist Occasional Paper, Office of the Chief Scientist, Canberra.

Roberts, M. E., Stewart, B. M., and Tingley, D. (2019). stm: R Package for Structural Topic Models. *Journal of Statistical Software 91*, pages 1–40.

Rogers, E. M. (2010). *Diffusion of Innovations, 4th Edition.* Simon and Schuster.

Saha, R. K., Saha, A. K., and Perry, D. E. (2013). Toward understanding the causes of unanswered questions in software information sites: a case study of stack overflow. In *9th Joint Meeting on Foundations of Software Engineering*, pages 663–666.

Schwab, K. (2017). *The Fourth Industrial Revolution.* Crown.

Scott, C. C. and Fensham, P. J. (1977). Student and teacher perceptions of the degree of difficulty in a mathematics course. *International Journal of Mathematical Education in Science and Technology*, 8(4):375–384.

SKBF and CSRE (2018). Rapport éducation [White paper].

Stevens, D., Totaro, M., and Zhu, Z. (2011). Assessing it Critical Skills and Revising the Mis Curriculum. *Journal of Computer Information Systems*, 51(3):85–95.

Strack, R., Kaufman, E., Ádám, K., Sigelman, M., Restuccia, D., and Taska, B. (2020). What's Trending in Jobs and Skills [White paper]. Technical report, Boston Consulting Group & Burning Glass Technologies.

Szabó, I. and Neusch, G. (2015). Dynamic Skill Gap Analysis Using Ontology Matching. In Kő, A. and Francesconi, E., editors, *Electronic Government and the Information Systems Perspective*, pages 231–242. Springer International Publishing.

Tambe, P. (2014). Big Data Investment, Skills, and Firm Value. *Management Science*, 60(6):1452–1469.

Tambe, P. and Hitt, L. M. (2012). The Productivity of Information Technology Investments: New Evidence from IT Labor Data. *Information Systems Research*, 23(3-part-1):599–617.

Tambe, P. and Hitt, L. M. (2013). Job Hopping, Information Technology Spillovers, and Productivity Growth. *Management Science*, 60(2):338–355.

Udemy (2020). 2020 Workplace Learning Trends Report: The Skills of the Future [White paper].

Vasilescu, B., Filkov, V., and Serebrenik, A. (2013). StackOverflow and GitHub: Associations between Software Development and Crowdsourced Knowledge. In *2013 International Conference on Social Computing*, pages 188–195.

Vassileva, J., McCalla, G. I., and Greer, J. E. (2016). From Small Seeds Grow Fruitful Trees: How the PHelpS Peer Help System Stimulated a Diverse and Innovative Research Agenda over 15 Years. *International Journal of Artificial Intelligence in Education*, 26(1):431–447.

Wai, L. (2016). Data science at Udemy: Agile experimentation with algorithms. In *2016 Future Technologies Conference (FTC)*, pages 355–360.

Wall, J. and Knapp, J. (2014). Learning computing topics in undergraduate information systems courses: managing perceived difficulty. *Journal of Information Systems Education*, 25(3):245–259.

Wallach, H. M., Mimno, D., and McCallum, A. (2009). Rethinking LDA: Why Priors Matter. In *Advances in neural information processing systems 22*, pages 1973–1981.

Wang, S., Chen, T.-H., and Hassan, A. E. (2018). Understanding the factors for fast answers in technical Q&A websites. *Empirical Software Engineering*, 23(3):1552–1593.

Whitehill, J., Bartlett, M., and Movellan, J. (2008). Measuring the Perceived Difficulty of a Lecture Using Automatic Facial Expression Recognition. In Woolf, B. P., Aïmeur, E., Nkambou, R., and Lajoie, S., editors, *Intelligent Tutoring Systems*, pages 668–670. Springer.

Williamson, B. (2017). *Big Data in Education: The Digital Future of Learning, Policy and Practice.* SAGE.

Yang, X.-L., Lo, D., Xia, X., Wan, Z.-Y., and Sun, J.-L. (2016). What Security Questions Do Developers Ask? A Large-Scale Study of Stack Overflow Posts. *Journal of Computer Science and Technology*, 31(5):910–924.

Zhu, M., Sari, A., and Lee, M. M. (2018). A systematic review of research methods and topics of the empirical MOOC literature (2014–2016). *The Internet and Higher Education*, 37:31–39.

Ramtin Yazdanian
Route des Flumeaux 1, 1008
Prilly

Phone : +41 76 7684962
Website: https://chili.epfl.ch/
https://people.epfl.ch/ramtin.yazdanian
E-mail : ramtin.yazdanian@gmail.com

Education

September 2017 – January 2022

PhD in Computer and Communications Sciences EPFL, Lausanne, Switzerland

Thesis title: Detecting Latent Training Needs Using Large Datasets
Thesis supervisor: Pierre Dillenbourg (supervisor), Robert West (co-supervisor)

September 2012 – July 2017

Bachelor of Science in Computer Engineering, Software Sharif University of Technology, Tehran, Iran

Thesis title: An Attribute Learning Method for Zero-Shot Learning.
Thesis supervisor: Mahdieh Soleymani Baghshah

Research projects

July 2020 – February 2021

Research Internship National University of Singapore (Remote)

Under the supervision of Prof. **Min Yen Kan**, and in collaboration with the national firm **SkillsFuture Singapore**, developed a machine learning pipeline that uses job ad trends to predict **"emerging skills"**: lower-popularity skills that are going to experience a surge in hiring demand in the near future. Successfully tested the method using data from the Singaporean ICT sector, and analysed its performance.

Publications

- R. Yazdanian, RL Davis, X Guo, F Lim, P Dillenbourg, MY Kan. On the Radar: Predicting Near-future Surges in Skill Demand to Provide Early Warning to Educators (Working title), Submitted to Computers and Education: Artificial Intelligence, publication pending. (2021)
- R Yazdanian, R West, P Dillenbourg. Keeping Up with the Trends: Analyzing the Dynamics of Online Learning and Hiring Platforms in the Software Programming Domain. International Journal of Artificial Intelligence in Education. (2020)

- R Yazdanian, L Zia, J Morgan, B Mansurov, R West. Eliciting New Wikipedia Users' Interests via Automatically Mined Questionnaires: For a Warm Welcome, Not a Cold Start. Proceedings of the International AAAI Conference on Web and Social Media 13. (2019)

- R Yazdanian, L Zia, R West. The Elicitation of New Users' Interests on Wikipedia. Wiki Workshop 2018

- R Yazdanian, SM Shojaee, MS Baghshah. An attribute learning method for zero-shot recognition. Proceedings of the 2017 Iranian Conference on Electrical Engineering (ICEE), 2235-2240 (2017).

Teaching

Teaching assistant for courses

- Applied Data Analysis
  Homework and exam design, teaching, office hours

- Lab in Data Science
  Teaching, office hours

- Software engineering
  Office hours

Supervisor for semester projects

- The Stack Overflow Annual Survey: A Look into the Future?
  Thanh Loïc Nguyen, Maxime Lemarignier, 2019

Awards and distinctions

- Outstanding Problem Solution Award at ICWSM 2019

  For the paper "Eliciting New Wikipedia Users' Interests via Automatically Mined Questionnaires: For a Warm Welcome, Not a Cold Start"

Languages

- English - C2

- Persian - Mother language

- French - B1

- Russian - A1

Personal information

- Date of Birth: 17.06.1994
- Nationality: Iranian
- Civil status: Married
- Children: None

Personal profile

Hardworking, motivated, and fast-learning data scientist. Coming from an engineering background, I aim to build a career as a data scientist / machine learning engineer. I love practical challenges that involve not only analyzing data and devising analytic and predictive pipelines, but also building those pipelines into practical solutions. While I am self-motivated, I feel most at home in a team environment. I have considerable experience analyzing data and building predictive pipelines in Python and handling large amounts of data using Apache Spark, and I am familiar with Git, SQL, and Java.