



# Keeping Up with the Trends: Analyzing the Dynamics of Online Learning and Hiring Platforms in the Software Programming Domain

Ramtin Yazdanian<sup>1</sup> · Robert West<sup>1</sup> · Pierre Dillenbourg<sup>1</sup>

Accepted: 12 November 2020 / Published online: 08 December 2020

© The Author(s) 2020

## Abstract

The Fourth Industrial Revolution has considerably sped up the pace of skill changes in many professional domains, with scores of new skills emerging and many old skills moving towards obsolescence. For these domains, identifying the new necessary skills in a timely manner is a difficult task, where existing methods are inadequate. Understanding the process, by which these new skills and technologies appear in and diffuse through a professional domain, could give training providers more time to identify these new skills and react. For this purpose, in the present work, we look at the dynamics between online learning platforms and online hiring platforms in the software programming profession, a rapidly evolving domain. To do so, we fuse four data sources together: Stack Overflow, an online community questions and answers (Q&A) platform; Google Trends, which provides online search trends from Google; Udemy, a platform offering skill-based Massively Open Online Courses (MOOCs) where anyone can create courses; and Stack Overflow Jobs, a job ad platform. We place these platforms along two axes: i) how much expertise it takes, on average, to create content on them, and ii) whether, in general, the decision to create content on them is made by individuals or by groups. Our results show that the topics under study have a systematic tendency to appear earlier on platforms where content creation requires (on average) less expertise and is done more individually, rather than by groups: Stack Overflow is found to be more agile than Udemy, which is itself more agile than Stack Overflow Jobs (Google Trends did not prove usable due to extreme data sparsity). However, our results also show that this tendency is not present for all new skills, and that the software programming profession as a whole is remarkably agile: there are usually only a few months between the first Stack Overflow appearance of a new topic, and its first appearance on Udemy or Stack Overflow Jobs.

---

This article belongs to the Topical Collection: *A festschrift in honour of Jim Greer*  
Guest Editors: Gord McCalla and Julita Vassileva

✉ Ramtin Yazdanian  
ramtin.yazdanian@epfl.ch

Extended author information available on the last page of the article.

In addition, we find that UdeMy's agility has dramatically increased over time. Our novel methodology is able to provide valuable insights into the dynamics between online education and job ad platforms, enabling training program creators to look at said dynamics for various topics and to understand the pace of change. This allows them to maintain better awareness of the trends and to prioritize their attention, both on the right topics and on the right platforms.

**Keywords** Web mining · MOOCs · Stack overflow · Job advertisements · Training needs analysis

## Introduction

The evolution of AI in education can be described along multiple dimensions. The dimension addressed in this paper concerns the scale of the processes being modelled by AI. From this viewpoint, AIED history can be depicted as a sequence of concentric circles. Initially, AI algorithms aimed at adapting instruction to individual learners, which requires inferring an accurate model of the learners' skills and knowledge. Since Jim Greer has been a pioneer of AIED research, it is not surprising that a key part of his contributions have been about learner modelling (McCalla et al. 2000; Zapata-Rivera and Greer 2000). Later on, this initial circle (computer + learner) has been broadened to include multiple learners (collaborative learning), then interactions with the whole class and the teacher(s), then the peers' community up to massive learner cohorts (MOOCs). The work of Jim Greer on finding helping peers (Greer et al. 1998a, b) illustrates these broader circles. The work presented here addresses yet a broader circle: society. It expands AIED beyond the frontiers of education systems, by modelling how transformations in markets and society do indeed affect education: the economy generates new training needs or skill requirements which reshape the activities of formal and informal education systems. In a nutshell, while most research in AIED uses AI to support the acquisition of new skills, the present contribution attempts to use statistical methods and AI to determine which skills should be acquired.

This societal extension of AIED is nonetheless a faithful pursuit of the long running AIED quest to develop adaptive systems: while AI has helped adapt education to learners, can it help adapt education to the economy and society as a whole? This question has become a priority, as education is challenged by the rapid changes in society such as the digital revolution, which have made many professional domains change very rapidly. The skills required for an individual to keep up with (and benefit from) changes tend to evolve much faster than educational curricula. In a short time span, many skills that used to be essential stand to become obsolete, with new skills taking their place. Therefore, many organizations and governments around the world are making considerable investments into *up-skilling* their workforce (Kim 2002). The first step to speed up this adaptation process would be identifying those emerging skills as early as possible. We believe that Jim would have appreciated the use of AI to make education more agile. This question concerns all professions, from bakers

and carpenters to software developers and the manufacturing industry, but we investigate it in the context of the software industry, as it is a fast-changing industry where there is much more data to analyze.

Many AIED scholars have already acknowledged the need for lifelong learning and the necessity of staying up-to-date on the newest skill trends in the labor market (Field 2006; Latchem et al. 2016). As such, Jim's extensive work on supporting lifelong learning has paved our way, in particular when it comes to the fact that a peer-help system such as PHelpS or iHelp (Vassileva et al. 2016) (that can help support lifelong learning) needs up-to-date skill hierarchies. The emergence of new skills in any domain should lead to adapted curricula, and this curricular adaptation will benefit from Jim's work on knowledge hierarchies (Vassileva et al. 2016) and tools for data-driven curricular changes. The vision that underlies our contribution is perfectly expressed in this sentence by Jim: "Academics pride themselves on evidence-informed decision making, but when it comes to making changes in their teaching practices, curricula, or academic programs, data and evidence seem to hold little sway." (Greer et al. 2016a, b). Our goal is to be able to guide these decision makers - in a data-driven fashion - towards the changes they may need to make to their curricula, making sure that they can keep up with societal and economic changes.

In existing literature, the process of finding the skills that a workforce needs to learn is known as Training Needs Analysis (TNA for short) (Anderson 1994). Much of the literature on TNA revolves around questionnaires that ask the employees of an organization (or of a group of organizations) to evaluate their level of mastery in previously identified skills that are *known to be important* to their performance (Anderson 1994). Such methods are too slow to react to rapid changes, and are inappropriate for emerging skills, since the skills assessed by such a questionnaire need to have already been identified. As a result, our goal would be to develop methodologies better suited to emerging skills and to innovations (which could be skills or technologies), the main drivers of the Fourth Industrial Revolution (Schwab 2016). According to the theory of "innovation diffusion" (Rogers 1995) - which provides a general theoretical framework for the process, through which innovations are adopted by firms in a professional domain - the adoption of an innovation by an agile vanguard and then by early adopters precede its wider adoption (if it does achieve wider adoption). Therefore, in order to stay ahead of the trends, understanding these early stages is paramount. If we can identify either the early adopters, or the early signs that foretell an innovation's later, more widespread adoption, then we can identify emerging skills much more rapidly than existing methods can. Even if we are unable to identify such early signs, understanding and quantifying how quickly the professional domain and its various sub-parts evolve would inform training program creators of the time frames involved and allow them to focus their attention on the most important areas.

In this paper, our 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, an immensely popular online Q&A platform for software developers;

2. Google Trends, a tool that provides the normalized Google search volume of a given query over time;
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.

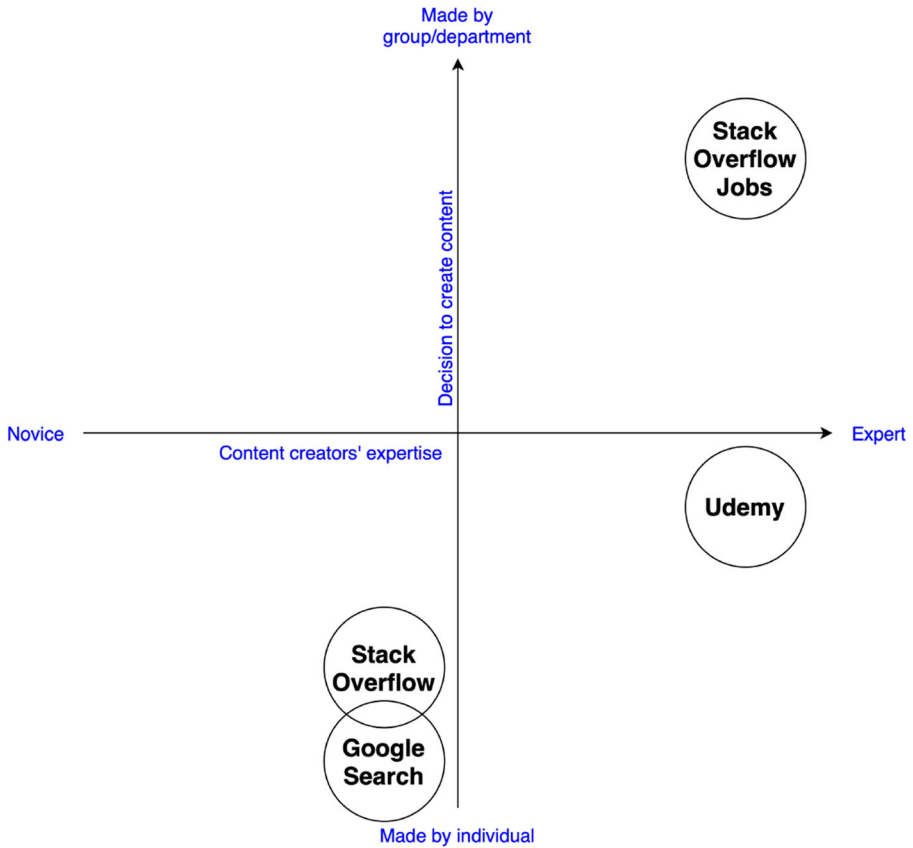
The new skills and technologies in our study are identified through (and represented by) Stack Overflow “tags”: user-created words or phrases that are used to describe the topics of Stack Overflow questions. At the time of this paper’s writing, tens of new tags are created on Stack Overflow every week<sup>1</sup>, 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, we analyze and compare the times at which each new tag shows up on each of the four platforms for the first time. We hypothesize 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<sup>2</sup> (or, in other words, when novices and experts alike can create content), and decisions to create content being made more individually, rather than by groups or departments. We place our four platforms along the two axes of “content creator expertise” and “individuality of content creation decision making”, as shown in Fig. 1, and we investigate 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 our study. We then hypothesize 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 our study.

Our results show that in the majority of cases, our 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. We also find the software programming profession to be 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. We also observe some variance in this delay and in the proportion of tags where our 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

---

<sup>1</sup>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.

<sup>2</sup>We should clarify here that by “creating content” for a platform like Google Trends, we mean simply searching for a query and thus generating data. For Udemy and Stack Overflow Jobs, it is posting a course/lecture or a job ad, respectively; while for Stack Overflow, we limit our definition of content creation to posting questions, for reasons that will be explained in the Methodology section.



**Fig. 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

or a framework, etc.)<sup>3</sup>. Regarding our other, prediction-related hypothesis, we find that signals based only on user activity on the more agile platforms appear to be insufficient for predicting the appearance of new topics on the less agile platforms. Our results 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. Our 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

<sup>3</sup>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 our work.

for expansion, allowing it to serve as a basis for deeper dives into a particular professional domain, especially software programming. Therefore, we believe that our work, with its focus on helping training providers in adapting their curricula to the changing skill landscape, is an important continuation and extension of Jim's work on the use of learning analytics for curricular adaptation (Greer et al. 2016a; Greer et al. 2016b) and can help expand the scope of educational research to include its economic context.

## Related Work

### Technological Change, Big Data, and Lifelong Learning

Today, innovative technologies are being developed at a rapid pace. With the fourth industrial revolution in full swing, many industries have been substantially disrupted – especially by ICT skills (Michaels et al. 2014) – and many fields (such as data science or cloud computing) owe their creation to this revolution (Schwab 2016). An innovation, be it the creation of a new product, or the creation of a new process for production (Gopalakrishnan and Damanpour 1997), can (and often does) bring with itself new skills that the workforce needs to learn (Kim 2002), and as a result, life-long learning has emerged as a crucial element of the new economy (Field 2006). Many entities, ranging from corporations to educational institutions to governments, react to this need for upskilling the workforce, as evidenced by the significant body of recent technical reports that analyze the skill needs of the present and the future (Prinsley and Baranyai 2015; Point 2016; Strack et al. 2019; LinkedIn Talent Solutions 2019; Coursera 2019). As stated before, the formal process for identifying the skills that need to be taught and the areas where training programs are needed is called Training Needs Analysis (TNA), and its traditional methods are ill-suited to rapidly changing environments, as they rely on interviews or questionnaires that ask employees about already-identified skills (Anderson 1994; Chiu et al. 1999; Iqbal and Khan 2011). This is where labor market analytics using big data come in.

### The role of Big Data

The belief that big data will serve as a transformative force in the economy is prevalent (Horton and Tambe 2015; Einav and Levin 2014). Horton and Tambe (Horton and Tambe 2015) discuss the emergence of online data sources from labor market intermediaries, such as online hiring platforms (i.e. job ad websites), Q&A forums, and online course platforms; these data sources provide new opportunities for empirical research on labor markets, and on skills in particular. 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. The former, 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, alongside many others (Horton and Tambe 2015). Among said platforms, the educational platforms are geared towards lifelong learning, and they especially enable informal e-learning (Latchem et al. 2016), although technologies such as Q&A forums can also be used to support more formal learning (Hammond 2005). 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). When it comes to making economic predictions (which, as mentioned, is one of our aims), 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.

### **Big Data Sources: Online Hiring and Learning Platforms**

Now, having explained the importance of big data in analyzing labor markets, we should discuss the types of online platforms that we believe would be suitable for our study (particularly platforms used for learning), before focusing on the data sources we have actually chosen for the study, which are online job ad platforms/collections, online Q&A platforms, and MOOC platforms<sup>4</sup>.

Many different kinds of digital software and platforms (many of them online) have been used to support lifelong learning, and they can be divided into two groups: purpose-built educational software, and software that was not designed for education but can be used for it. The first group includes MOOC platforms such as Coursera, edX, Udacity, and Udemy (Buhl and Andreasen 2018), 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 (Kimmons and Lifelong Learning 2018; 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 videoconferenc-

---

<sup>4</sup>These three types of data sources correspond to Stack Overflow Jobs, Stack Overflow, and Udemy, respectively; Google Trends is absent from this discussion because its importance is indirect and dependent on Stack Overflow's, as we will see in the Methodology section.

ing tools which allow people to communicate - peer-to-peer and over long distances - for educational matters (Kimmons and Lifelong Learning 2018). The ones that we are interested in are those that are very widely used (so as not to limit the scale of the study, which is on an entire professional domain) and whose data is not sensitive to the point of being unobtainable. This means that the best-suited educational data sources for a study like ours would be MOOC platforms, public Q&A platforms, blogs, and social networks with publicly available data (such as Twitter), although for domains where data from peer help systems or personal knowledge managers is available, those could also serve as valuable sources.

In the rest of this section, we will look at the literature on online hiring platforms, Q&A platforms, and MOOC platforms, and we will be particularly interested in studies that conduct analyses of trends in technologies and/or skills on the three types of platforms that we consider most relevant to a study such as ours<sup>5</sup>.

### 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) (Point 2016; Strack et al. 2019; LinkedIn Talent Solutions 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. 2019; LinkedIn Talent Solutions 2019) are often thorough in analyzing various types of skills. In particular, Burning Glass Technologies (Strack et al. 2019) distinguishes 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

---

<sup>5</sup>In addition, it is worth noting that all the data in our study are publicly available, thus boosting the reproducibility of our work.



and their trends (Barua et al. 2012; Allamanis et al. 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. 2012; Johri and Bansal 2018; Ishola and McCalla 2016). Some of these (Barua et al. 2012; Johri and Bansal 2018) 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 Blie 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.

Here, it is worth noting that the annual Stack Overflow Developer Survey<sup>6</sup> provides data on the technologies used by developers, although the data comes from a survey and not from the Q&A forums themselves.

## 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 for Business 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

---

<sup>6</sup><https://insights.stackoverflow.com/survey/>

et al. 2016). Studies such as Coursera (2019) and (Udemy for Business 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 what we had for 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.

## Positioning our Work

In our literature review, we have sketched the landscape of research efforts that are relevant for our research endeavor, as summarized in Table 1. We 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 we explore 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. Our originality is to explore 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). We explore the relationship between platforms 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. Our paper is a first step in this direction.

## Research Methodology

### Platforms and datasets

As mentioned before, the data we analyze 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 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

**Table 1** A summary of the relevant state of the art and existing gaps in the literature on the platforms that we have deemed useful for our study

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”

The reasons why we have only looked at these three types of platforms are detailed in “[Big Data Sources: Online Hiring and Learning Platforms](#)”

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<sup>7</sup>. We are primarily interested in 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 can help us see two indicators of interest we cannot see on 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<sup>8</sup> to Google Trends. In this dataset, we are interested in the search volumes over time for various topics. As we lack access to raw Google search data, and since Google’s method for computing the query counts is not public, the amount of interpretation we can do 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 that we have considered in our study.
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. We 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<sup>9</sup>, openly avail-

---

<sup>7</sup><https://archive.org/download/stackexchange>

<sup>8</sup><https://github.com/GeneralMills/pytrends>

<sup>9</sup><https://www.udemy.com/developers/affiliate/>

able data on Udemy courses may be obtained. The data we are interested in 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 Stack Overflow tags that we have mentioned before, 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<sup>10</sup>, which is the web scraping bot of The Internet Archive. In this dataset, we are interested in the title and description of each ad, its posting date, and its tags.

These four data sources, summarized in Table 2, provide different perspectives into the topics that exist in the software programming domain (of which we are mainly interested in the new topics). 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 Fig. 1, we place these four platforms 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) incline us to believe 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. The position we have chosen is not fully accurate, but for the purposes of our study, we do not need greater accuracy, as we effectively never compare Google Trends and Stack Overflow (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).

<sup>10</sup>[https://web.archive.org/web/\\*/stackoverflow.com/jobs](https://web.archive.org/web/*/stackoverflow.com/jobs)

**Table 2** A summary of the relevant data in the data sources used in our study, along with the earliest date for which data can be obtained from each data source

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

- 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 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.

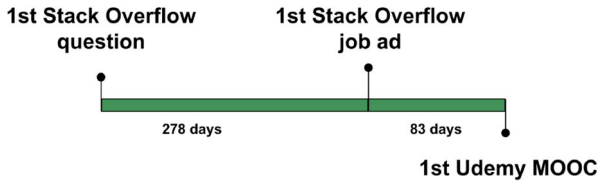
## Aims and Methods

So far, we have talked about topics (which are our proxies for skills, which would be training needs) and how the four platforms provide different views on them, and in this part, we will discuss our general methodology for identifying and studying those topics, and we will discuss how the aims of our study shape our methodology.

## Aims

Our goal is twofold. Firstly, we aim to understand the early life (and in particular, the first appearance) of new topics on our four platforms, looking at how agile each platform is, how the platforms evolve over time, and how differently they behave with respect to different topics. Secondly, we aim 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. We call the first goal our **descriptive** aim and the second our **predictive** aim, and we will refer to these two clusters of platforms as expert/group-driven and novice/individual-driven, respectively, although we will also engage in individual discussions of the platforms. In order to work towards these two aims, we will 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. 2 shows the tag `vuejs2` (for the Javascript framework `Vue.js`, version 2), having appeared first on Stack Overflow, then on Stack 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



**Fig. 2** The delays between appearance on Stack Overflow, Stack Overflow Jobs, and Udemey for the tag `vuejs2`, which is about Vue.js version 2, a Javascript framework

preempt the expert/group-driven platforms, would have for creating a course or lecture in the ideal case – the case where we could predict any topic’s eventual importance correctly, right when it appeared on a novice/individual-driven platform. We use multiple criteria for the prediction window in order to also gauge the strength of the predictive signals. Fig. 2 shows the prediction windows for `vuejs2` for Stack Overflow Jobs (278 days) and for Udemey (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. Our investigation of the early life of new topics on our 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, we have discussed what we aim to analyze about new “topics”, and now it is time to discuss what we have chosen 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 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, we believe that Stack Overflow tags are a good proxy for the skills whose dynamics we aim to analyze. We acknowledge that this approach mostly excludes higher order skills such as ‘agile development practices’ or ‘redesigning a client-server architecture for minimizing bandwidth requirements’.

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.

**Table 3** Examples of Stack Overflow questions and their tags for four questions

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

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



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 our “descriptive” and “predictive” goals 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 we want 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 we can observe a wider range of tag behaviors. 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, we would never learn about the topic’s existence. This issue is therefore very difficult to address, but given Stack Overflow’s popularity, we believe that a new topic never appearing on Stack Overflow is very unlikely, and we will be studying a long period of time. Again, this second issue is more of a problem for the predictive aim, because as we mentioned before, for a tag that does not appear on novice/individual-driven platforms before the expert/group-driven platforms, our prediction task is meaningless. As a result, we will not be able to address these two issues for the predictive aim. For the descriptive aim, however, we will come back to this issue later in this section and introduce a partial solution, in the form of tag **popularity**, helping us make our results more robust.

Before we proceed further, we should discuss why we have chosen Stack Overflow tags as the representatives of our topics, 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. We have three reasons for doing so:

1. Tags are user-created and also edited by moderators. Therefore, a great amount of crowdsourced manual work has gone into them, and using tags allows us to utilize this already-done work, rather than performing topic modelling from scratch, which would require us to find the optimal number of topics, and then to interpret the topics and figure out what each one really means. Interpreting a topic model is a well-known and important problem that requires 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.

### Connecting the datasets

In order to track the same topics across the four data sources, for each tag we perform the following:

- We observe its popularity on Stack Overflow in the form of question counts and vote count time series, in particular looking at the earliest question that has the tag.
- We retrieve its Google Trends search volumes.
- We find its occurrences in the lectures/course titles of Udemy MOOCs.
- We find it in the descriptions and tags of Stack Overflow Jobs ads.

In order to find occurrences of each Stack Overflow tag in the titles of Udemy course syllabi and in job ad descriptions, our solution is simply to find 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 would lead to a considerable error potential. This is why we have opted for the simpler, direct matching approach. The details of our matching of tags to their occurrences in text are found in Appendix A. After performing the matching, we eliminate ‘generic’ tags such as “introduction” from our list of tags. These tags are neither technologies nor concepts, and therefore are not topics that we would be interested in. 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, we noticed that most of these tags appear on Udemy and Stack Overflow Jobs during the early phase of the platforms’ existence, and thus we can remove them by eliminating all the tags that have appeared on these platforms during their early life. This brings us to our discussion of warm-up periods.

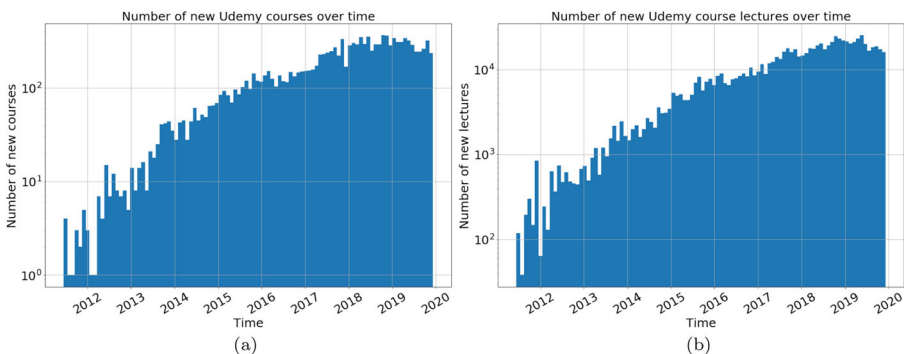
### Starting dates and platform warm-up

An important question in our work is the periods that we aim to 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 our 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

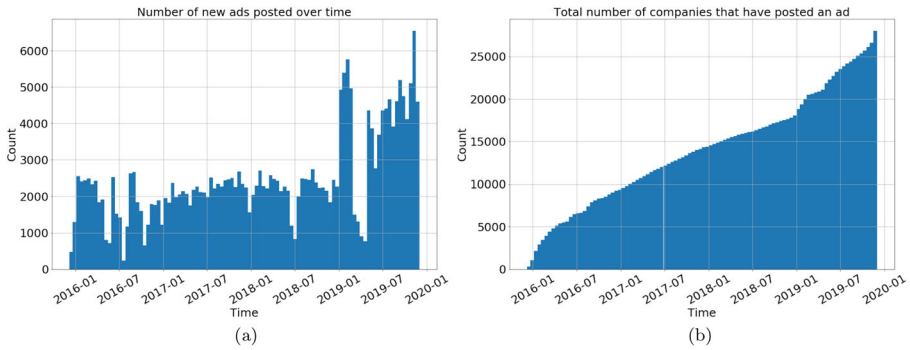
need arising from the emergence of a technology or skill, but are rather a sign that people are starting to recognize and use *the platform itself*.

Figure 3 shows histograms of new course and new course lecture counts on Udemy from the beginning until December 2019. Similarly, Fig. 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. We identify the end of the warm-up periods of these two platforms by a stabilization of the number of new tags appearing on each platform. Figure 5 shows a histogram of new tag counts over time for both platforms. We have fixed the end of the warm-up period for Udemy at July 2013 (which is right after the jump seen in Fig. 3a) and for Stack Overflow Jobs at October 2016 (which is again after a jump, seen in Fig. 4a). In order to have fair comparisons between the different datasets, whenever we connect several datasets together, we choose a **starting date** that is after the end of every dataset's warm-up period, and filter out all the tags that have appeared on *any* of the platforms before that date. This has the added advantage of removing the generic tags that mostly appear during a platform's early life.

In addition, as we can see in Fig. 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, until the present), we have decided to limit our study 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 we are studying the time at which a tag manifests on each dataset for the first time. We simply only look at the data that has been generated up until the end of 2018.



**Fig. 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

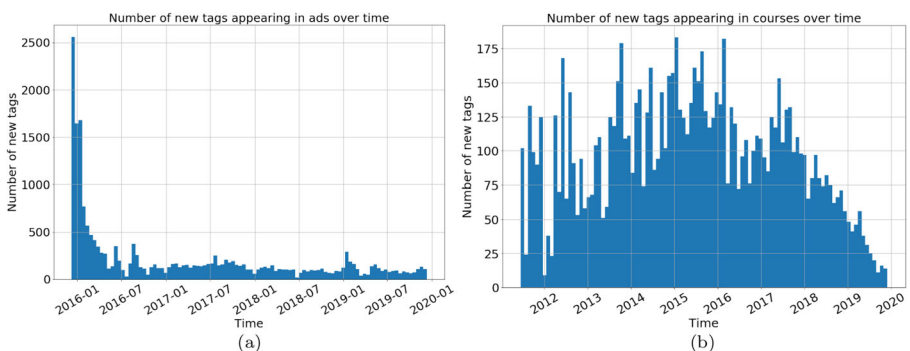


**Fig. 4** Number of new ads published (a) and the total number of companies that have posted an ad so far (b) on Stack Overflow Jobs over time (until Dec. 2019), divided into 100 bins. Each bar is the number of new job ads published during that time bin's duration. The total number of unique companies at the beginning of January 2019 is 18095

### “Popular” tags

When we introduced tags as the topics in our study, we mentioned that there are two situations where our argument that “new tags represent new topics” is incorrect. Those issues are harder to address for our predictive aim, as this aim requires quick action and does not give us much time to wait for community action on the tags. For the descriptive aim, however, we have more time, and can thus address these issues using signals from the community. We will now introduce our concept of “popular” tags.

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 allow us to double check whether this issue would invalidate our hypotheses. To this end, we will define “popular” tags, and then report our descriptive measures separately both for “all” tags and for “popular” tags only.



**Fig. 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

To define “popular” tags, since we are interested in the early life of a tag and not its entire lifetime, we compute two measures for 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 allow us to know 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 our studied period (i.e. the end of 2018). We define popular tags 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<sup>11</sup>. The medians for our two popularity measures are 32 questions per year and 32 votes per year, and our criteria give us a total of 5575 popular tags. Of course, this definition is not set in stone, because we have no ground truth on popularity per se, but we believe that our combination of measures, and our two analyses (popular tags and all tags) will give us a reasonably accurate picture of the different behaviors of tags.

## Research questions

Summing up our discussions of our methodology and aims, and based on the definitions we have offered earlier, we define our research questions 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?
- **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?

## Results

### Appearance Ordering

Our 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.

---

<sup>11</sup>These two dates are the warm-up date of Udemy and the end date of our 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 that we our study.

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<sup>12</sup>.

Most tags trivially appear on Stack Overflow, but their existence on the all other three platforms is not a given. This discriminates **dataset groups**, where in each group, we only consider those tags 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 that we gave 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 sometimes gets quite small.

Table 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"<sup>13</sup> and "% SO first" gives us the percentage of tags in this group that appeared on Stack Overflow before appearing on Udemy. Since the hypothesis we wish to investigate for RQ1 is that tags have a tendency to appear on more novice/individual-driven platforms before more expert/group-driven platforms, we have conducted a statistical test 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).

We see several important observations in Table 4 regarding **RQ1**:

1. **The null hypothesis of our statistical test 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 to 6), but it becomes weakly rejected (for popular tags) or unrejected (for all matched tags) when we match all of our datasets 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 drops considerably when all the

<sup>12</sup>In order to retrieve the specific tag on Google Trends, we treat the tag as an n-gram: we replace the "-"es with whitespace and then put quotations around the query, in order to make sure we are not including unrelated search queries through individual words in the query.

<sup>13</sup>In our tables, SO is Stack Overflow and GT is Google Trends.

**Table 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

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

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 our statistical test (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

datasets are matched together. In addition, we can see that **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 we match 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 we 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 our descriptive aim, this lends greater support to our hypothesis.

2. **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 our 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.
3. **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, we see that 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”.
4. **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 we investigated, Google Trends was very rarely the first place where a new tag appeared, being surpassed by every other dataset by a large margin. We have shown only one example in Table 4, with other examples skipped for brevity. An investigation of the reasons behind this revealed that many of the tags we investigated had a volume of *zero* on Google Trends, due to their extreme specificity. Part of this is because we have used 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 we do not know how exactly Google Trends handles queries with multiple words (e.g. 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, we have excluded Google Trends for this very reason.

To summarize the results of this section, we conclude that the hypothesis in our RQ1 is 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 that we see 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 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.

## Delays and Prediction Windows

To answer our second research question, we will look at summary statistics (1st quartile, median, 3rd quartile) for the prediction windows of tags in various dataset-starting date groups, shown in Table 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, we also look at the time between the 1st and 5th Stack Overflow questions of each tag. 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 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

**Table 5** Examples of popular and non-popular tags that appeared on Stack Overflow, Udemy, or Stack Overflow Jobs first

	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

Popular tags are in bold as established in the previous table

**Table 6** Number of matched tags, and the three quartiles of event delays (in days) for different dataset-starting date groups

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	<b>731</b>	q1: 36 q2: 84 q3: 224	<b>731</b>	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	<b>117</b>	q1: 18 q2: 41 q3: 88	<b>117</b>	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	<b>167</b>	q1: 20 q2: 46 q3: 89	-	-	<b>167</b>	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

Table 6 (continued)

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, SO Jobs, Oct. 2016, popular	<b>57</b>	q1: 15 q2: 39 q3: 75	<b>57</b>	q1: -24 q2: 125 q3: 300	<b>57</b>	q1: 13 q2: 237 q3: 489	<b>57</b>	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	-	-	-	-	-	-	<b>206</b>	q1: -147 q2: 44 q3: 213

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

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, we can observe an interesting pattern: 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, our 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 we have greater evidence of importance, and therefore, they provide more reliable information when it comes to our aim of **describing** the behavior of the platforms in our 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 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, we first 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 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, we have 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, we create two sets of tags:

1. Tags created on both Stack Overflow and Udemy between July 2013 and October 2015.
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 us avoid potential biases arising from one set covering a longer period of time, and thus having a greater likelihood of having larger delays<sup>14</sup>. Statistics on the delays can be seen in Table 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, we use the Mann-Whitney U test. The null hypothesis of the test is that the delay of a tag randomly sampled from one set is

<sup>14</sup>This is because we can clearly see that the delay distribution is biased towards positive delay values.

**Table 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

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	<b>127</b>	q1: 35 q2: 79 q3: 239	<b>127</b>	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	<b>117</b>	q1: 18 q2: 41 q3: 88	<b>117</b>	q1: 21 q2: 141 q3: 303

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, implying that it is not the Stack Overflow community becoming less agile, but rather the Udemy creators' community becoming more agile

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. We do not know whether this increased agility is 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), leading us to mainly give credit to the expanded creator and student communities for Udemy's increased agility.

### Stronger signals of rising importance

Tables 6 and 7 tell us 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 we have is 3 to 4 months. As a result, our predictive aim now begs an important question: what other signals do we 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, we have calculated two sets of measures for each of the two expert/group-driven platforms (we have moved the plots 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<sup>15</sup> 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 B), 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 give us a very small prediction window.

---

<sup>15</sup>The first week where all of the tag's questions combined get at least N votes.

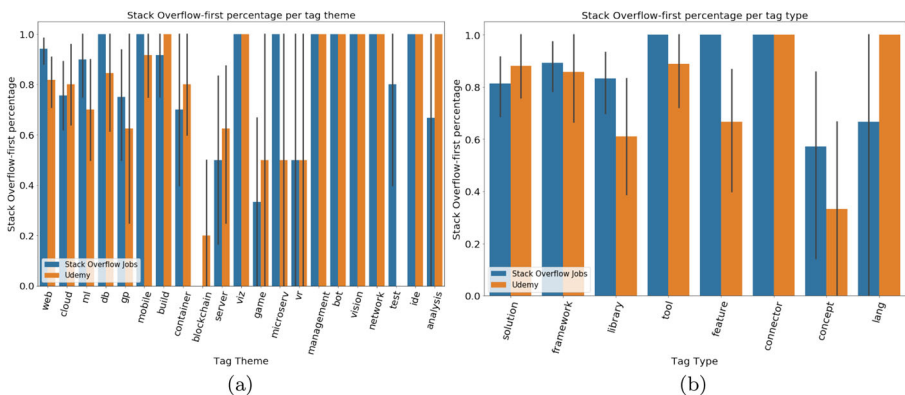
## Tag themes and types

As mentioned before, our study is not complete without an analysis of the **content** of the tags that we are studying. In order to investigate the differences in appearance orderings and prediction windows between various tags, we first had to choose a set of tags, and then 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, we have chosen the “popular” post-October 2016 tags that appeared either on Stack Overflow Jobs *or* on Udemy — we have chosen the popular tags over all tags in order to reduce the size of the tag set considerably. This gives us a total of 227 tags. We performed the annotation of each tag 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 C.

We are mainly interested in two sets of metrics:

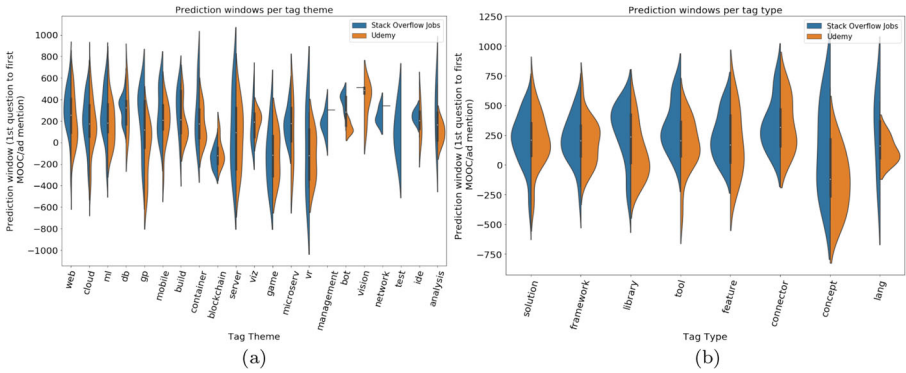
1. The proportion (from 0 to 1) of Stack Overflow-first tags among tags of each theme and of each type, and
2. The distribution of the prediction windows (i.e. time to first appearance on Udemy/Stack Overflow Jobs) among tags of each theme and of each type.

Figure 6 shows bar charts of the former (with error bars), while Fig. 7 shows violin plots 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 we can see, the different types and themes are not so strongly



**Fig. 6** Bar charts (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)





**Fig. 7** 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

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 C), 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<sup>16</sup>.
- 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

<sup>16</sup>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.

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 we gain 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.

## **Discussion**

### **Implications of Results**

#### **Implications of descriptive findings**

The descriptive aspect of our study serves the purpose of establishing a broad (while reasonably deep) view of an entire professional domain, allowing us to understand 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. Our descriptive findings have allowed us to quantify the agility of the software programming domain. In addition, we have proved that the platforms that are more novice/individual-driven can give us 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 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 intuitive, but the details that we have shown in our paper, 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 us to prescribe a prioritization of platforms for people who need to understand emerging skill trends. Our 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 we would put Udemy rather than Stack Overflow Jobs second 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 we have seen, we can summarize our prediction-related findings as the following:

- The appearance orderings we have 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 our prediction task.
- The prediction windows we have 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 we have calculated, are signals that are also present for many of the tags that do not appear on any expert/group-driven platform.

The conclusion we draw from these is that our prediction 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 we have ruled out, specifically, is predicting the appearance of the first course/lecture or job ad using user activity data on Stack Overflow. The prediction of further interest in the topic (through more courses, lectures or ads) remains unexplored 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, our prediction task 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.

## Generalizability

An important question when it comes to our methodology 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, our methodology is extensible to other domains as long as 1) online hiring and educational platforms exist for that domain, and 2) we have a way to detect those new topics in that domain. In our study, Stack Overflow tags served this 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<sup>17</sup> 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. we would not expect to see large Q&A forums on operating industrial machinery. 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.

## Threats to validity

In the methodology section, we discussed several caveats that arose from decisions we had to make in designing our methodology. Here, we will discuss how those caveats, and others, could affect the validity of the conclusions that we have drawn from our data.

---

<sup>17</sup><https://insights.stackoverflow.com/survey/2019>

- 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), we believe the impact of these caveats to be limited, and we believe that our classification of these two as more novice-driven compared to Udemy and Stack Overflow Jobs is in any case fully accurate.
- Only a small subset of the tags get matched between all our datasets. However, we have also analyzed Stack Overflow with each of the expert/group-driven datasets separately, and when it comes to the first two research questions, the results are similar for these larger sets of matched tags. Therefore, this does not present a considerable threat to our findings.
- In general, we may have missed some tag matches, given that we look for an exact match in the syllabus of a Udemy course or in the description of a job ad. As a result, we most likely lack 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 our methodology section, when introducing tags, we mentioned the two potential issues: that a tag might not represent a new or important topic, and that a new topic might never appear as a tag. As discussed back then, we believe the latter to be a minor issue as Stack Overflow is a very popular Q&A website, and we have addressed the former issue by analyzing popular tags separately, and we have a similar confirmation of our hypothesis for them (and the hypothesis is actually confirmed more strongly for the popular tags), and these two issues are therefore not a large threat to the validity of our study.
- Our 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 we get all of our tags from, we have 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 our 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.

## Conclusions and Future Work

We have presented a methodology for analyzing the dynamics of new topics among online educational platforms and hiring platforms in the software programming domain. Our 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. We have also quantified the impressive agility of the software programming domain, demonstrating that it can be indeed very difficult to predict the digital traces of the earliest adopters of a new skill or technology. Our 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, our 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.

Our work lends itself to multiple directions for future work. The most straightforward direction, which we have already discussed, is to apply our methodology to another professional domain, which would also allow us to compare different domains. A second direction is to analyze the **spread** of new topics on the platforms in our 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 our work is to include other, less open and more centralized MOOC platforms, and even more traditional educational institutions such as universities in the study, which were not included in our study in order to keep the scope reasonable and because Udemy, with its unique properties, was (in our opinion) the most interesting MOOC platform to study.

**Acknowledgments** We would like to thank the Swiss State Secretariat for Education, Research, and Innovation (SEFRI) for their generous funding of this project. Also, we would like to thank Stack Overflow for giving back to the community and publicly releasing their data, and The Internet Archive for making such massive amounts of historical internet data available for free.

**Funding** Open access funding provided by EPFL Lausanne

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## Matching Stack Overflow tags to their occurrences in text

In our matching of tags to their occurrences in Udemy lecture and course titles and Stack Overflow Jobs ad descriptions, we had to pay attention to several details in order to increase recall. First of all, Stack Overflow tags are not necessarily unigrams, and therefore, in order to match text to tags, we need to generate n-grams. We use a Porter stemmer on both the text and to each unigram in every tag. We thereby avoid false negatives due to grammatical features (such as a word being plural in the text and singular in the tag). We also took extra care 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 occur to be 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. We have dealt with this problem in two stages. First, we have used two heuristics 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.

After narrowing down the possible pairs of unrelated tag-subtag pairs, we manually go through the pairs and only keep those that are actually unrelated, so that we can later filter them out as we see them<sup>18</sup>.

## Appendix B: Prediction windows for stronger signals of tag importance

This appendix (Figs. 8 and 9) includes 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.

## Appendix C: Tag themes and types

This appendix provides definitions for tag themes and types in Table 8. 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. Figures 6 and 7 respectively show box plots of the Stack Overflow-first proportions and violin plots of the prediction windows for various tag themes and types (in each figure, sub-figure (a) is for themes and (b) is

---

<sup>18</sup>When such a tag and subtag occur in a text, we only keep the longer tag and do not match the subtag

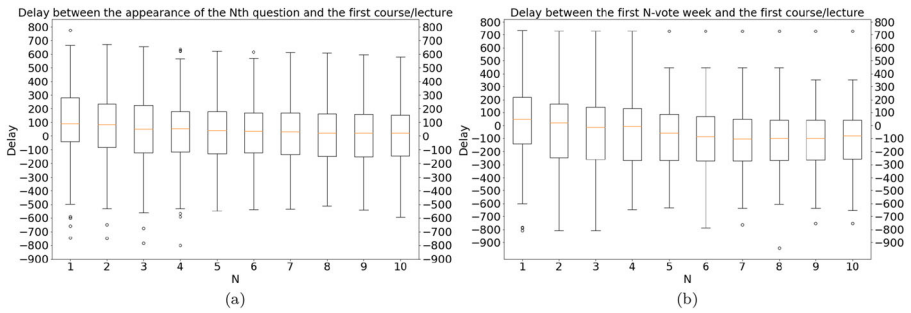
**Table 8** Tag themes and types that appear at least 5 times in our set of 227 tags, along with the number of tags they are related to and a short description of what they stand for

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
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 Relationship 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.

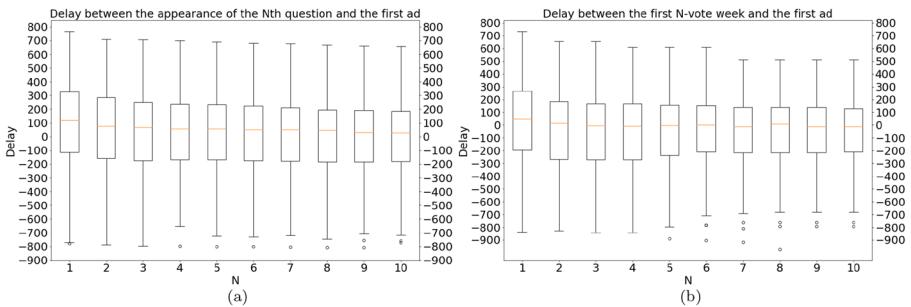


**Table 8** (continued)

Name	Theme or Type?	# of tags	Description
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).



**Fig. 8** 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)



**Fig. 9** 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

## References

- Allamanis, M., Sutton, C., Why, W. (2013). What Analyzing Stack Overflow Questions by Topic, Type, and Code. *Proceedings of the 10th Working Conference on Mining Software Repositories*.
- Anderson, G. (1994). A proactive model for training needs analysis. *Journal of European Industrial Training*, 18, 23–28.
- Anderson, A., Huttenlocher, D., Kleinberg, J., Leskovec, J. (2012). Discovering Value from Community Activity on Focused Question Answering Sites: A Case Study of Stack Overflow. *Proceedings of the 18th ACM SIGKDD international conference on Knowledge Discovery and Data Mining* (pp. 850–858).
- Barua, A., Thomas, S.W., Hassan, A.E. (2012). What are developers talking about? an analysis of topics and trends in stack overflow. *Empirical Software Engineering*, 19, 619–654.
- Bhat, V., Gokhale, A., Jadhav, R., Pudipeddi, J., Akoglu, L. (2014). Min(e)d Your Tags: Analysis of Question Response Time in StackOverflow. *2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 328–335.
- Blei, D.M., Ng, A.Y., Jordan, M.I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, 3, 993–1022.
- Bozkurt, A., Keskin, N.O., de Waard, I. (2016). Research trends in massive open online course (MOOC) theses and dissertations: Surfing the tsunami wave. *Open Praxis*, 8(3), 203–221.
- Buhl, M., & Andreasen, L.B. (2018). Learning Potentials and Educational Challenges of Massive Open Online Courses (MOOCs) in Lifelong Learning. *International Review of Education*, 64, 151–160.
- Chang, J., Gerrish, S., Wang, C., Boyd-Graber, J.L., Blei, D.M. (2009). Reading tea leaves: How humans interpret topic models. *Advances in Neural Information Processing Systems (NIPS)*, 22, 288–296.
- Chiu, W., Thompson, D., Mak, W., Lo, K.L. (1999). Re-thinking Training Needs Analysis: A Proposed Framework for Literature Review. *Personnel Review*, 28(1/2), 77–90.
- Conache, M., Dima, R., Mutu, A. (2016). A comparative analysis of MOOC (massive open online course) platforms. *Informatica Economica*, 20, 5–14.
- Correa, D., & Sureka, A. (2014). Chaff from the wheat: Characterization and Modeling of Deleted Questions on Stack Overflow. *Proceedings of the 23rd International Conference on World Wide Web*.
- Coursera (2019). Global skills index 2019 technical report.
- Ebben, M., & Murphy, J.S. (2014). Unpacking MOOC scholarly discourse: a review of nascent MOOC scholarship. *Learning, Media, and Technology*, 39, 328–345.
- Einav, L., & Levin, J. (2014). The data revolution and economic analysis. *Innovation Policy and the Economy*, 14, 1–24.
- Estelles, E., Del Moral, E., 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.
- Field, J. (2006). Lifelong learning and the new educational order (second revised edition). Stoke-on-Trent, Trentham Books.
- Gopalakrishnan, S., & Damanpour, F. (1997). A review of innovation research in economics, sociology, and technology management. *Omega*, 25, 15–28.
- Greer, J., McCalla, G., Collins, J.A., Kumar, V., Meagher, P., Vassileva, J. (1998a). Supporting peer help and collaboration in distributed workplace environments. *International Journal of Artificial Intelligence in Education (IJAIED)*, 9, 159–177.
- Greer, J., McCalla, G., Cooke, J., Collins, J., Kumar, V., Bishop, A., Vassileva, J. (1998b). The intelligent helpdesk: Supporting Peer-Help in a university course. *International Conference on Intelligent Tutoring Systems (ITS)*, 1998, 494–503.
- Greer, J.E., Molinaro, M., Ochoa, X., McKay, T. (2016a). Learning analytics for curriculum and program quality improvement (PCLA 2016). *Sixth International Conference on Learning Analytics & Knowledge (LAK)*, 494–495.
- Greer, J.E., Thompson, C., Banow, R., Frost, S. (2016b). Data-Driven Programmatic Change at Universities: What Works and How. *1st Learning Analytics for Curriculum and Program Quality Improvement Workshop*, 30–33.

- Hammond, M. (2005). A review of recent papers on online discussion in teaching and learning in higher education. *Journal of Asynchronous Learning Networks*, 9(3), 9–23.
- Hiranrat, C., & Harncharnchai, A. (2018). Using text mining to discover skills demanded in software development jobs in thailand. *Proceedings of the 2nd International Conference on Education and Multimedia Technology*, 112–116.
- Horton, J.J., & Tambe, P. (2015). Labor economists get their microscope: Big data and labor market analysis. *Big Data*, 3, 130–137.
- Iqbal, M.Z., & Khan, R.A. (2011). The growing concept and uses of training needs assessment: a review with proposed model. *Journal of European Industrial Training*, 35(5), 439–466.
- Ishola, O.M., & McCalla, G. (2016). Detecting and supporting the evolving knowledge interests of lifelong professionals. *EC-TEL 2016: Adaptive and Adaptable Learning*, 595–599.
- Johri, V., & Bansal, S. (2018). Identifying Trends in Technologies and Programming Languages Using Topic Modeling. *12th IEEE International Conference on Semantic Computing*, 391–396.
- Kim, Y. (2002). A state of art review on the impact of technology on skill demand in OECD countries. *Journal of Education and Work*, 15(1).
- Kimmons, R., & Lifelong Learning (2018). In Ottenbreit-Leftwich, A., & Kimmons, R. (Eds.) *The K-12 Educational Technology Handbook*: EdTech Books.
- Latchem, C., Technology, L.earning., Informal, L.lifelong. (2016). Learning Technology and Lifelong Informal, Self-directed, and Non-formal Learning. *The Wiley Handbook of Learning Technology*, 180–199.
- LinkedIn Talent Solutions (2019). Future of skills 2019 technical report.
- McCalla, G., Vassileva, J., Greer, J., Bull, S. (2000). Active learner modelling. *International Conference on Intelligent Tutoring Systems (ITS)*, 2000, 53–62.
- Michaels, G., Natraj, A., 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.
- Papoutsoglou, M., Mittas, N., Angelis, L. (2017). Mining People Analytics from StackOverflow Job Advertisements. *43rd Euromicro Conference on Software Engineering and Advanced Applications*, 108–115.
- Point, B.eyond. (2016). Click the expanding demand for coding skills, Burning glass technologies and oracle academy.
- Ponzanelli, L., Mocchi, A., Bacchelli, A., Lanza, M., Fullerton, D. (2014). Improving low quality stack overflow post detection. *Proceedings of the IEEE international conference on software maintenance and evolution*.
- Prinsley, R., & 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. Technical Report. *Occasional Paper Series*, 9.
- Rogers, E.M. (1995). *Diffusion of innovations*, 4th edn. New York: The Free Press.
- Schwab, K. (2016). *The Fourth Industrial Revolution*, 1st edn. New York: Crown Business.
- Strack, R., Kaufman, E., Kotsis, A., Sigelman, M., Restuccia, D., Taska, B. (2019). *What's trending in jobs and skills*. Technical report, Burning Glass Technologies.
- Tambe, P., & Hitt, L.M. (2012). The productivity of information technology investments: New evidence from IT labor data. *Information systems research*, 23.
- Tambe, P., & Hitt, L.M. (2013). Job Hopping, Information Technology Spillovers, and Productivity Growth. *Management science*, 60.
- Tambe, P. (2014). Big Data Investment, Skills, and Firm Value. *Management science*, 60.
- Udemy for Business (2020). 2020 Workplace Learning Trends Report: The skills of the future technical report.
- Vasilescu, B., Filkov, V., Serebrenik, A. (2013). Stackoverflow and GitHub: Associations between Software Development and Crowdsourced Knowledge. *Proceedings of the International Conference on Social Computing*.
- Vassileva, J., McCalla, G.I., 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, 431–447.
- Wai, L. (2016). Data science at udemy: Agile experimentation with algorithms. *Future Technologies Conference*, 355–360.

- Wallach, H.M., Mimno, D.M., McCallum, A., Rethinking, L. D. A. (2009). Why priors matter. *Advances in Neural Information Processing Systems (NIPS)*, 22, 1973–1981.
- Zapata-Rivera, J.D., & Greer, J. (2000). Inspecting and visualizing distributed bayesian student models. *International Conference on Intelligent Tutoring Systems (ITS)*, 2000, 544–553.
- Zhu, M., Sari, A., Lee, M.M. (2018). A systematic review of research methods and topics of the empirical MOOC literature (2014–2016). *International Journal of The Internet and Higher Education* (37), 31–39.

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

## Affiliations

Ramtin Yazdanian<sup>1</sup> · Robert West<sup>1</sup> · Pierre Dillenbourg<sup>1</sup>

Robert West  
robert.west@epfl.ch

Pierre Dillenbourg  
pierre.dillenbourg@epfl.ch

<sup>1</sup> Ecole Polytechnique Federale de Lausanne (EPFL), Lausanne, Switzerland