

# Teaching Technology to (Future) Lawyers

Mikołaj Barczentewicz\*

## Abstract

The article offers a reflection on how applications of computer technology (including data analytics) are and may be taught to (future) lawyers and what are the benefits and limitations of the different approaches. There is a growing sense among legal professionals and law teachers that the technological changes in the practice of law are likely to promote the kind of knowledge and skills that law graduates often do not possess today. Teaching computer technology can be done in various ways and at various depths, and those different ways and levels have different cost and benefit considerations. The article discusses four models of teaching technology: (1) teaching basic technological literacy, (2) more advanced but general technology teaching, (3) teaching computer programming and quantitative methods and (4) teaching a particular aspect of technology – other than programming (e.g. cybersecurity). I suggest that there are strong reasons for all current and future lawyers to acquire proficiency in effective uses of office and legal research software and standard means of online communication and basic cybersecurity. This can be combined with teaching of numerical and informational literacy. I also claim that advanced technology topics, like computer programming, should be taught only to the extent that this is justified by the direct need for such skills and knowledge in students' future careers, which I predict to be true for only a minority of current lawyers and law students.

**Keywords:** legal education, law and technology, legal analytics, technology education, technological literacy

## 1 Introduction<sup>1</sup>

It is widely accepted and, I think, true that technology – or, more specifically, computers and computer networks – will be playing an even greater role in the future of legal practice than it is today.<sup>2</sup> Perhaps the changes are not going to be very radical in the short term. However, some specialised software tools will likely continue to transform certain aspects of legal practice. Among the

obvious examples are e-discovery (and due diligence),<sup>3</sup> basic legal research<sup>4</sup> and contract automation tools,<sup>5</sup> which though sometimes not precise or 'self-driving' enough for many practical contexts today, are likely to keep improving. The same is true with regard to artificial intelligence algorithms aiming to predict how a court (or another authority) would apply the law in a given factual scenario: there is a lot of promise, but the results are still somewhat underwhelming (for instance, a great deal of manual work is still required to create such tools).<sup>6</sup> There is a growing sense among legal professionals and law teachers that the technological changes in the practice of law are likely to promote the kind of knowledge and skills that law graduates often do not possess today.<sup>7</sup> Different kinds of technological

3. E-discovery may mean both 'the process by which computers search a database for keywords that lawyers agree are marks of relevance' and the more advanced 'predictive coding' where 'lawyers look at a sample of the larger set of documents' and '[c]omputer technicians help construct algorithms that predict whether a document is relevant'; J.O. McGinnis and R.G. Pearce, 'The Great Disruption: How Machine Intelligence Will Transform the Role of Lawyers in the Delivery of Legal Services Colloquium: The Legal Profession's Monopoly on the Practice of Law', 82 *Fordham Law Review* 3041, at 3047 (2013).
4. The improvements include, for instance, 'semantic search' (which 'will allow lawyers to input natural language queries to computers, and the computers will respond semantically to those queries with directly relevant information'), machine judgments on strength of precedent, and automatic detection of the most relevant cases based, e.g., on a scan of the text of a court brief; *ibid.*, at 3048-3050.
5. According to McGinnis and Pearce: 'In the future, machine processing will be able to automate a form, tailor it according to the specific facts and legal arguments, and track its effect in future litigation. As hardware and software capacity improves, so too will the generated documents. We predict that within ten to fifteen years, computer-based services will routinely generate the first draft of most transactional documents'; *ibid.*, at 3050.
6. See e.g. *ibid.*, at 3052-3053; D.M. Katz, M.J. Bommarito II & J. Blackman, 'A General Approach for Predicting the Behavior of the Supreme Court of the United States', 12 *PLoS ONE* (2017), at <https://doi.org/10.1371/journal.pone.0174698>; M. Medvedeva, M. Vols & M. Wieling, 'Using Machine Learning to Predict Decisions of the European Court of Human Rights', 28 *Artificial Intelligence and Law* 237 (2020); M. Barczentewicz, 'Combining AI and Digitization of Judgments for Access to Justice', in S. Chishti and others (eds.), *The LegalTech Book: The Legal Technology Handbook for Investors, Entrepreneurs and FinTech Visionaries* (2020).
7. See, e.g., M. Pistone and J.J. Hoeffner, 'No Path But One: Law School Survival in an Age of Disruptive Technology', 59 *Wayne Law Review* 193 (2013); R.W. Staudt, 'Introduction, Justice, Lawyering and Legal Education in the Digital Age', 88 *Chicago-Kent Law Review* 687 (2013); T. Rostain, R. Skalbeck & K.G. Mulcahy, 'Thinking Like a Lawyer, Designing Like and Architect: Preparing Students for the 21st Century Practice', 88 *Chicago-Kent Law Review* 743 (2013); D.M. Katz, 'The MIT School of Law? A Perspective on Legal Education in the 21st Century', *University of Illinois Law Review* 1431 (2014); M. Pistone, 'Law Schools and Technology: Where We Are and Where We Are Heading', 64 *Journal of Legal Education* 586 (2015); V. Janeček, R. Williams & E. Keep, 'Education for the Provision of Technologically Enhanced Legal Services' 40 *Computer Law & Security Review* 105519

\* Mikołaj Barczentewicz is the Research Director, Surrey Law and Technology Hub, as well as Senior Lecturer (Associate Professor) in Law, University of Surrey School of Law. He is also a Research Associate of the University of Oxford Centre for Technology and Global Affairs. I would like to express my thanks to the anonymous referees.

1. This article develops a short piece published as M. Barczentewicz, 'Learn in Code', *Solicitors Journal* (February 2020).  
 2. J. Armour, R. Parnham & M. Sako, 'Augmented Lawyering', European Corporate Governance Institute (ECGI) – Law Working Paper 5582020, 2020, at <http://dx.doi.org/10.2139/ssrn.3688896>.

expertise are cited as examples of what the lawyers of the future will need to know and be able to do. For instance, Daniel Katz argued that a law school of the future (the MIT School of Law) should have a ‘curriculum that strongly emphasize[s] science, technology, process engineering, predictive analytics, and mathematical and computational modelling’.<sup>8</sup> In this article, I reflect on how technology is and may be taught to (future) lawyers and what benefits and limitations are associated with the different approaches. This article is not intended as a comprehensive survey of technology education in law schools and law faculties. The examples I discuss are meant to be illustrative.

I am concerned here solely with computer technology, not with technology in the broader sense. Also, I do not consider the issue of teaching technology law and regulation but only that of acquiring skills and knowledge in technology as such. Furthermore, I do not dispute that the growing importance of computer technology may call for lawyers to acquire business, design or process management skills, which do not fit into the narrower conception of technology skills and knowledge on which I focus in this article.<sup>9</sup>

The crucial preliminary question for any discussion of teaching technology to lawyers and law students is the extent to which lawyers today and in the near future will benefit from such knowledge and skills. In a survey of English solicitors reported by Janeček, Williams and Keep, ‘90 per cent of English solicitors indicated that they would need some training concerning AI and digital technology in the next three years’.<sup>10</sup> What is more, 71 per cent of respondents expected data analytics to be a training need.<sup>11</sup> It must be noted that virtually all of the lawyers surveyed most likely have but a very vague idea of what data analytics is. Hence, for many, the expectation that they will need training in data analytics is probably an expression of the willingness to jump on the popular bandwagon, not an informed judgment. Armour, Parnham and Sako are clearly right in predicting that some lawyers will in the near future be consumers of technology, with their jobs augmented by technology but not in a way that requires any detailed knowledge of it.<sup>12</sup> True, some lawyers or law graduates will be involved in the development of legal technology and will need various levels of technological expertise (some – though probably not many – may even benefit from full interdisciplinarity, for instance, as both lawyers and software engineers).<sup>13</sup> Moreover, as today,

some lawyers will benefit from technological expertise in their legal practice (especially in areas like intellectual property or, more broadly, technology regulation). The big question is the proportions among those types of future lawyering.

### 1.1 Three Kinds of Lawyers

In other words, there are three levels of need for technological literacy for current and future lawyers:

1. those who benefit significantly only from basic proficiency (but greater than the current norm),
2. those who benefit significantly from more advanced knowledge about the behind-the-scenes workings of technology (just like planning and construction lawyers benefit from knowledge about architecture and construction technology),
3. those who benefit significantly from even more advanced technological proficiency, including advanced *practical* skills such as computer programming.

My prediction is that the vast majority of future lawyers will stay at levels (1) and (2). Lawyers remain consumers of computer technology and need to know only as much about it as their field of law requires (more in intellectual property, much less in family law). In other words, I do not expect the current situation to change dramatically. Software, unlike traditional in-person legal advice, is easily scalable, and hence a relatively small number of producers can service a very large number of consumers (in this case lawyers or those who seek legal advice). Despite attempts in a growing number of law firms today,<sup>14</sup> I do not expect that in-house software development will be as significant for legal practice as simply licensing software from comparatively few providers (of course, some law firms may become software providers themselves or spin off such companies). Moreover, optimists about technological improvement who expect software to be increasingly better at automating some legal tasks should not forget that software development itself is also likely to be positively affected by this trend.<sup>15</sup> It would be curiously myopic to think that automation means that ‘everyone’ should learn computer programming as it is practised now. The same (or very similar) technological improvements as the ones that spearhead automation in legal practice are likely to do the same in software development, and to do so even faster.

Even if thinking only about the most immediate future, before any further significant technological improvements, there are good reasons for lawyers and law students to be cautious about investing too much effort in their technology education. For instance, computer programming (or more broadly, software engineering) is easy to do poorly but difficult to do well. The risks asso-

(2021); R. Williams, V. Janeček & E. Keep, ‘What Is the Role of Law Schools in the 21st Century?’, 2020; A. Smith and N. Spencer, ‘Do Lawyers Need to Learn to Code? A Practitioner Perspective on the “Polytechnic” Future of Legal Education’, in C. Denvir (ed.), *Modernising Legal Education* (2020); Armour, Parnham & Sako, above n. 2; Barczeniewicz, above n. 1.

8. Katz, above n. 7, at 1465.

9. See, e.g., Smith and Spencer, above n. 7; Williams, Janeček & Keep, above n. 7.

10. Janeček, Williams & Keep, above n. 7, at 4.

11. *Ibid.*

12. Armour, Parnham & Sako, above n. 2, at 56-7.

13. *Ibid.*, at 57-8.

14. *Ibid.*, at 35-41.

15. See, e.g., B.W. Sorte, P.P. Joshi & V. Jagtap, ‘Use of Artificial Intelligence in Software Development Life Cycle: A State of the Art Review’, 3 *International Journal of Advanced Engineering and Global Technology* 398 (2015); M. Barenkamp, J. Rebstadt & O. Thomas, ‘Applications of AI in Classical Software Engineering’, 2 *AI Perspectives* 1 (2020).

ciated with bad code are very significant, especially in terms of reliability and security. The benefits of division of labour, i.e. letting the coders do the coding, are overwhelming in most circumstances. This is not to deny that relatively few lawyers can make significant contributions to the development of legal technology as fully fledged technology experts, even as computer programmers. It is only important not to lose a sense of proportion over how many people will be in that group and thus on how much of that kind of training needs to be provided.

However, some degree of more advanced understanding of technology will remain crucial (just as it is now) to those lawyers, classified in my level (2) above, who need it in legal practice (intellectual property, technology law and regulation, and so on). Moreover, a small minority of lawyers (my level (3)) will benefit from even greater understanding and from advanced *practical* skills such as computer programming. Such lawyers may work as ‘legal engineers’ or ‘legal technologists’ participating in developing legal technology on a par with engineers or as members of multidisciplinary teams. They can also work as empirical legal researchers, particularly in academia or government (although it is likely that empirical research will require less programming in the future owing to the development of appropriate software).

It is a separate question of what proportion of lawyers would benefit from more basic training in issues such as cybersecurity, numerical literacy or even greater proficiency in using office and legal research software. Here the answer, I think, is that most lawyers (as with most professionals in other fields) should gain such fundamental proficiency. And this is true also for the youngest and future members of the profession. There is a popular myth that the young people today are ‘digital natives’ and that they are more tech-savvy than previous generations.<sup>16</sup> This claim is usually made in a very vague way, making it difficult to verify empirically. However, serious attempts to do so show that it is indeed a myth.<sup>17</sup> Those who grew up in the age of the internet may be more adept at clicking through interfaces of some software applications that they use daily, but their understanding of the behind-the-scenes mechanisms of computer technology is not at all impressive.<sup>18</sup> Characteristically, computer security habits of the younger people are just as bad as those of their elders.<sup>19</sup>

16. See, e.g., E.J. Helsper and R. Eynon, ‘Digital Natives: Where Is the Evidence?’, 36 *British Educational Research Journal* 503 (2010); D. Bates, ‘Are “Digital Natives” Equipped to Conquer the Legal Landscape?’, 13 *Legal Information Management* 172 (2013); T. Ståhl, ‘How ICT Savvy Are Digital Natives Actually?’, 12 *Nordic Journal of Digital Literacy* 89 (2017); T. Judd, ‘The Rise and Fall (?) Of the Digital Natives’, 34 *Australasian Journal of Educational Technology* 99 (2018).

17. Helsper and Eynon, above n. 16; Bates, above n. 16; Judd, above n. 16, at 99.

18. See, e.g., J. Fraillon and others, *IEA International Computer and Information Literacy Study 2018 Assessment Framework* (2019).

19. See, e.g., S.S. Tirumala, A. Sarrafzadeh & P. Pang, ‘A Survey on Internet Usage and Cybersecurity Awareness in Students’, 2016 *14th Annual Conference on Privacy, Security and Trust (PST)* (2016); J.D. Thompson, G.L. Herman, T. Scheponek, L. Oliva, A. Sherman, E. Golaszewski, D. Phatak, & K. Patsourakos, ‘Student Misconceptions about Cyberse-

Having broadly reflected on the scope of the need for technology I now turn to a discussion of four models of teaching about technology already present in law faculties. I will then return to the question of what kind of technology education is suitable depending on different career paths of lawyers and law graduates.

## 2 Models of Teaching about Technology

A number of universities and other education providers offer some form of technology education for undergraduate or postgraduate law students. In this part of the article, I discuss several examples of such courses and classify them in four groups, or ‘models’. In the first model basic technological and numerical literacy are taught. The remaining three models are concerned with more advanced technological proficiency. In the second model computer technology in general is taught in a more advanced way. The third model concerns teaching computer programming, and the fourth focuses on one specific aspect of computer technology (for instance, cybersecurity). My discussion is not meant as a comprehensive survey, and I do not claim that the examples chosen are the best in the world (because this would require a comprehensive comparison that I did not undertake), but I do consider them to be well designed.

### 2.1 Teaching Basic Technological Literacy

The first model of teaching technological proficiency is the most basic one and includes the teaching of effective uses of office and legal research software and standard means of online communication, cybersecurity. This may be paired with training in business skills, service and product design and process management, but I leave that issue aside as it is beyond the scope of this article.<sup>20</sup> Teaching basic technology skills may also be fruitfully connected to assisting (future) lawyers in gaining two other kinds of literacy: informational literacy and numerical literacy, both of which need improvement.

Law students and young lawyers may feel confident in their use of technology (to put it colloquially, they can google things very quickly), but this confidence quickly dissipates when faced with some even seemingly basic tasks required in the study and practice of law (at least in the United Kingdom, many court judgments cannot be found through Google, not to mention information such as what was the judgment’s subsequent authoritative treatment).<sup>21</sup> Probably all law schools offer students some form of an introduction to legal research and databases available to them, but this is likely to be limited to several hours in the ‘welcome week’ of the first year or a

curity Concepts: Analysis of Think-Aloud Interviews’, 2018 *Journal of Cybersecurity Education, Research and Practice* 5 (2018).

20. Janeček, Williams & Keep, above n. 7; Smith and Spencer, above n. 7.

21. See also Bates, above n. 16, at 176.

short online course. Experiences from Cambridge University described by Bates,<sup>22</sup> as well as my own observations, show worrying deficiencies in information literacy even among students in their later years. Similarly, students struggle with effective use of office software to complete tasks like formatting a document in Microsoft Word or using footnotes. This is not a reason to scoff as I am sure that proficiency in, for example, using Microsoft Word's 'styles' or numbering of paragraphs eludes many practising lawyers and academics.

One way this can be addressed is through entry-level modules on legal skills (legal research and writing) designed in a way that does not assume that young people require minimum (if any) instruction in effective use of software. An example of such successful effort in teaching proficiency in Microsoft Word at the University of Buffalo was described by Detweiler.<sup>23</sup> Another possible way is to include such instruction in core law subjects. For example, in the early weeks of a first-semester law module students could be provided with detailed narrated video tutorials (screen recordings) of how to research answers for the questions they are asked to prepare for those weeks.

Regarding numerical literacy, it is already a skill beneficial to all professionals and a necessity in some areas of law (financial regulation, tax law). With the propagation of artificial intelligence (machine learning) tools, there is a growing need for basic numerical literacy to include at least the fundamentals of statistics needed for informed use of products of machine learning. My own experience, which I am sure is widely shared, is that law schools cannot rely on their students' prior education in that respect. Hence, including numerical literacy in legal research modules or as stand-alone modules, perhaps delivered online, may be advisable. While discussing the other teaching models further on, I provide examples of how more advanced numeracy can be taught in 'legal analytics' or 'computational law' courses, often together with computer programming. I emphasise that the need for basic numeracy skills is broader than for programming skills, so it is likely suboptimal if the only teaching (or at least encouragement for independent learning) that a law student receives in this respect is at this advanced level, which is by and large unnecessary.

Finally, with regard to cybersecurity education, this is a particularly difficult issue, because unlike good research or writing skills, cybersecurity literacy for most people lacks immediate, easily perceived rewards. Cybersecurity skills, if practised, reduce – but never eliminate – one's risk of being a victim of a cyberattack. Moreover, cybersecurity can get very technical very quickly – as I show in Section 2.4 while discussing Yale Law School's 'Cybersecurity' course – but it is also not an obvious question to say how much cybersecurity training is too

little and how much is too much (for most people). On the other hand, lawyers, bound to protect the confidentiality of information about their clients or employers, constitute very attractive targets for attackers. For a lawyer to be a victim of a cyberattack, especially because of a kind of contributory negligence (e.g. in a phishing attack<sup>24</sup>), could (and often should) have very grave consequences. I am sceptical that a quick online cybersecurity module (especially in the initial weeks of one's studies or work) may have a meaningful effect on most people's security hygiene. One alternative way to approach cybersecurity education for lawyers and law students is through something like a system of fire drills. For instance, students could be regularly targeted with emails containing links that they should not click on (e.g. because of a suspicious originating domain), and if they do they would receive immediate feedback that they failed a cybersecurity drill with an invitation to online training showing them how to be more secure in the future. This kind of education can to largely be automated and delivered on a university (organisation) level.

I now turn to the three models of teaching more advanced technology topics.

## 2.2 More Advanced Teaching about Computer Technology in General

One way of providing more advanced teaching about computer technology is through a broad survey of salient topics, without singling out any particular topic. This has been the strategy of Oxford University's 'Law and Computer Science'<sup>25</sup> and Harvard University's 'CS50 for Lawyers'.<sup>26</sup> The two courses differ in their methods of learning and teaching. The Harvard course was designed to be delivered online to large numbers of students, whereas the Oxford course emphasises group work and instructor supervision (and thus enrolled only twelve postgraduate computer students in law and an equal number from computer science). The Oxford course is much more law oriented than the Harvard one and seems to cover more technology law than technology as such. Neither of the courses aims to teach students to program computer software on their own.

As reported by members of the teaching team, in its first year the Oxford course was structured in the following way:

The first half of the course focused on AI and digital technology in legal practice (the sphere primarily relevant to this paper); the second half of the course on

22. *Ibid.*, at 174-5.

23. B. Detweiler, 'A Quick Word About Technology Competence: The University at Buffalo School of Law's Microsoft Word Training Program', 25 *Perspectives: Teaching Legal Research and Writing* 97 (2017).

24. See, e.g., R. Dhamija, J.D. Tygar & M. Hearst, 'Why Phishing Works', *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Association for Computing Machinery 2006), at <https://doi.org/10.1145/1124772.1124861>.

25. Janeček, Williams & Keep, above n. 7. See also 'Law and Computer Science: 2019-2020', available at: [www.cs.ox.ac.uk/teaching/courses/2019-2020/LawandCS/](http://www.cs.ox.ac.uk/teaching/courses/2019-2020/LawandCS/), archived at: <https://perma.cc/CZE5-RCN9> (last visited 30 January 2021).

26. 'CS50 for Lawyers 2019', available at: <https://cs50.harvard.edu/law/2019/>, archived at: <https://perma.cc/ZBW7-D6TL> (last visited 30 January 2021).

questions of substantive law brought about by such technology (which as noted above, are beyond the scope of this paper). The experiential part of the course was based on a group project (each group containing three students from each discipline) that resulted in a pitch-like session to experts from the profession.<sup>27</sup>

The key idea behind the course design was ‘to explore how computer scientists and lawyers of the future will need to work together’.<sup>28</sup> This fits well with the adopted strategy of limited explicit instruction about technology combined with the focus on interdisciplinary group work and exchange of knowledge between students. The law students participating in the Oxford course may have ended up with a less comprehensive picture of computer technology than those who took the Harvard course, but it is possible that they gained skills that are likely to be more practically useful for them as lawyers who interact with engineers. However, purely on the basis of the synopsis of the course content and the relative lack of introductions to basic aspects of computer technology there, it would not be surprising if law students struggled with technological concepts and had to do a considerable amount of independent study. Perhaps supplementing the Oxford course with the kind of contents that are covered in the Harvard course might have made it more effective, at least for some students.

In contrast to the Oxford course, Harvard’s ‘CS50 for Lawyers’ explicitly covered issues like the basics of programming, algorithms, data structures, databases and cybersecurity.<sup>29</sup> The Harvard course even included some very basic programming tasks as at-home assignments. However, those tasks can be seen more as a familiarisation with the idea of programming than as teaching programming as a skill.

Simplifying matters a fair bit, one may be tempted to say that the Oxford course explicitly teaches lawyers to interact with engineers while leaving learning about technology more implicit (as a side effect of the interdisciplinary interactions and of explicit teaching about technology law), whereas the Harvard course teaches about technology explicitly and about interacting with technology specialists more implicitly. Neither of the courses turns lawyers into technology specialists and, in particular, equips them with sufficient knowledge and skills to develop software on their own, but given how unlikely it is that many lawyers today or in the near future will benefit from such in-depth grasp of technology, this is hardly a significant drawback. However, even the level of technology-related skills and knowledge that the two courses aim to provide is unnecessary for the vast majority of lawyers, while being very useful if not crucial for some small minority. Hence, there is a strong case for such technology courses to be offered as

optional for law students, but not as core modules in general legal education.

### 2.3 Teaching Computer Programming (Coding)

The third and often discussed model of teaching technology is hands-on teaching of computer programming (coding). Teaching of coding to law students may be done without any direct connection with the law – for example through an elective module delivered by a computer scientist and without any adjustments for law students.<sup>30</sup> However, it may also be embedded in a law-specific module, for example on ‘legal analytics’ or ‘Artificial Intelligence in law’.<sup>31</sup> A growing number of law schools offer modules with practical computer programming.<sup>32</sup> Those include modules offered within the Law and Technology Initiative at The University of Manchester,<sup>33</sup> the LLM in LegalTech at the University of Swansea,<sup>34</sup> ‘Applied Legal Data Analytics & AI’ course at the University of Pittsburgh School of Law,<sup>35</sup> ‘Introduction to Quantitative & Computational Legal Reasoning’ at the University of Iowa College of Law<sup>36</sup> and ‘Computational Law’ at the Stanford Law School (previously called ‘Legal informatics’).<sup>37</sup> Of those examples, only Iowa, Pittsburgh and Swansea state clearly that their teaching involves instruction in computer programming using a general-purpose programming language. In all three cases the language of choice is Python,<sup>38</sup> which is very popular in academic and commercial applications, especially in data science. Hence, the major benefit of learning Python is that it is a highly marketable skill in itself.<sup>39</sup> Naturally, the question remains as to the extent to which it is a marketable skill for law graduates in particular. As I noted in the previ-

27. Janeček, Williams & Keep, above n. 7.

28. ‘Law and Computer Science: 2019-2020’, above n. 25.

29. ‘CS50 for Lawyers 2019’, above n. 26.

30. Bocconi University in Milan offers such an elective course that is available to law postgraduates; see [www.unibocconi.eu/wps/wcm/connect/4d8d627d-9249-4710-887f-585c71a3c861/Scheda+Programming+with+Python.pdf?MOD=AJPERES&CVID=moqkLuG](http://www.unibocconi.eu/wps/wcm/connect/4d8d627d-9249-4710-887f-585c71a3c861/Scheda+Programming+with+Python.pdf?MOD=AJPERES&CVID=moqkLuG), archived at: <https://perma.cc/E5CP-88ZC> (last visited 30 January 2021).

31. K.D. Ashley, ‘Teaching Law and Digital Age Legal Practice with an AI and Law Seminar’, 88 *Chicago-Kent Law Review* 783 (2013).

32. See R. Tromans, ‘Legal Tech Courses’, *Artificial Lawyer*, available at: [www.artificiallawyer.com/legal-tech-courses/](http://www.artificiallawyer.com/legal-tech-courses/), archived at: <https://perma.cc/B2AU-RFCZ> (last visited 30 January 2021).

33. University of Manchester, ‘Law and Technology Initiative (LaTI)’, available at: [www.law.manchester.ac.uk/research/themes/law-money-technology/law-technology-initiative-2/](http://www.law.manchester.ac.uk/research/themes/law-money-technology/law-technology-initiative-2/), archived at: <https://perma.cc/N26Z-M765> (last visited 30 January 2021).

34. University of Swansea, ‘LLM in “LegalTech”’, available at: [www.swansea.ac.uk/postgraduate/taught/law/llmlegaltech/](http://www.swansea.ac.uk/postgraduate/taught/law/llmlegaltech/), archived at: <https://perma.cc/K66D-ZQXW> (last visited 30 January 2021).

35. University of Pittsburgh School of Law, ‘Applied Legal Data Analytics & AI’, available at: [www.law.pitt.edu/academics/courses/catalog/5719](http://www.law.pitt.edu/academics/courses/catalog/5719), archived at: <https://perma.cc/ZM2U-33PT>; <https://luimagroup.github.io/appliedlegalanalytics/>, archived at: <https://perma.cc/68QT-4ZLG> (last visited 30 January 2021).

36. P. Gowder, ‘Introduction to Quantitative & Computational Legal Reasoning’, available at: <https://sociologicalgobbledygook.com/>, archived at: <https://perma.cc/HNN8-L7GG> (last visited 30 January 2021).

37. Stanford Law School, ‘Computational Law’, available at: <https://law.stanford.edu/courses/computational-law/>, archived at: <https://perma.cc/Q4NQ-JY85>; <http://complaw.stanford.edu/>, archived at: <https://perma.cc/8CZP-RR9A> (last visited 30 January 2021).

38. G. Van Rossum and F.L. Drake, *Python 3 Reference Manual* (2009).

39. See Stack Overflow, ‘Developer Survey Results 2019’, available at: <https://insights.stackoverflow.com/survey/2019>, archived at: <https://perma.cc/W9UL-LTL6> (last visited 30 January 2021).

ous section of this article, arguably very few lawyers do and will benefit from being able to code. However, the minority who will benefit from it in the near future will likely benefit the most from knowing general-purpose languages such as Python or JavaScript.

The University of Pittsburgh's 'Applied Legal Data Analytics & AI' shares some of the teaching strategy with the Oxford course discussed in the previous subsection.<sup>40</sup> As in the Oxford course, Pittsburgh students are assessed through a project to be prepared in interdisciplinary groups meant to represent both legal and engineering competencies.<sup>41</sup> The main difference is that the Pittsburgh course requires all participants to complete practical programming and data analysis tasks. However, the course does not include instruction in the basics of programming in Python. To participate in this course, students are expected to either have a background in Python programming or learn the language on their own within the first few weeks of the course. The Pittsburgh course is law specific, but it is open to non-law students. Notably, their approach is focused on data analytics and thus covers an introduction to machine learning (and natural language processing in particular) as well as programmatic extraction and transformation of data originating from legal texts.

'Introduction to Quantitative & Computational Legal Reasoning' at the University of Iowa College of Law has similar teaching aims as the Pittsburgh course.<sup>42</sup> One key difference is in assessment through individually completed problem sets, not through group work. The Iowa course also explicitly covers the basics of programming in Python (which is a prerequisite at Pittsburgh). Similarly, in terms of statistics, the Iowa course focuses on the fundamentals with less time devoted to advanced topics like machine learning. The courses share a focus on practical training in programmatic data analysis of legal texts.

The emphasis on training in statistics and related skills (e.g. extracting data from texts, 'cleaning' the data), clear in both the Iowa and Pittsburgh courses, makes those courses particularly valuable. Students who complete those courses gain not only a capacity to code, which may or may not be of practical use to them in their future careers. They also gain significant numerical literacy, greater than in the first teaching model discussed in Section 2.1.

An additional benefit of learning data analytics and computational modelling, while focusing on case law or on legislation, is that it may help students gain a deeper understanding of the law. As Kevin Ashley argued, developing or reverse engineering computational models of law or legal reasoning forces us to make explicit many issues that lawyers tend to do intuitively and that stu-

dents are often expected to grasp without having them explained.<sup>43</sup>

The Stanford course took a noticeably different path and focused on teaching the fundamental concepts of computer programming on the example of a logic programming language, Epilog. Epilog is relatively unlikely to be used outside of a teaching context but is arguably well suited for learning of fundamental concepts of computer programming.<sup>44</sup> Epilog can be relatively easy for beginners to learn and use because it follows the syntax of symbolic logic. However, this benefit comes at the opportunity cost of not familiarising students with a different style of general-purpose programming that dominates academic and commercial uses, represented, for example, by Python. To some extent, learning the basic concepts of programming in Epilog should make it easier to start learning a language like Python. However, it is debatable whether choosing a language like Epilog over a language like Python is adequately beneficial even for beginners, especially given that Python skills are much more directly applicable outside of the teaching context. Moreover, the Stanford course does not cover quantitative methods of the kind the Iowa and Pittsburgh courses focus on. This may also contribute to the course being relatively easier for students than the other courses discussed here but, again, at the cost of more direct practical relevance.

A seemingly similar approach, but one that is actually very different from teaching to code, is to enable students to create 'apps' with the use of software tools that do not require programming skills in any of the general-purpose programming languages, or even a language like Epilog. Instead, such tools offer graphical interfaces ('no-code') and simple quasi-programming languages (a kind of 'low-code').<sup>45</sup> Depending on the software platform used, teaching assisted with such tools may potentially have similar benefits to the approach adopted by the Stanford course. That is, it may familiarise students with the basic concepts of programming like data structures and the logic of algorithms. This benefit comes with the limitations discussed in the Stanford case. However, it is also possible that the adopted no-code platform will be so simplified and 'user-friendly' that the students using it will not learn even the basic concepts of programming. At worst, the students may just learn 'how to click' through a particular interface of a particular piece of (soon-to-be-obsolete) software while gaining very limited transferable technology skills. Naturally, a course adopting this approach could deliver other learning outcomes than acquiring hard technology knowledge and skills, so limited value from the technological perspective does not necessarily mean that this is never a worthwhile teaching method. I do not discuss this possibility further as my concern in this article is exclusively on teaching technology.

40. See also Ashley, above n. 31.

41. 'Applied Legal Analytics & AI: Spring 2019', available at: <https://luimagroup.github.io/appliedlegalanalytics/>, archived at: <https://perma.cc/68QT-4ZLG> (last visited 30 January 2021).

42. Gowder, above n. 36.

43. Ashley, above n. 31, at 787-788.

44. 'Epilog', available at: <http://epilog.stanford.edu/>, archived at: <https://perma.cc/Z5WV-P74U> (last visited 30 January 2021).

45. Rostain, Skalbeck & Mulcahy, above n. 7, at 745.

## 2.4 Teaching a Particular Aspect of Technology – Other Than Programming

The fourth model of teaching about computer technology that I distinguish is advanced teaching of some specific aspect of technology, other than general-purpose programming. This could be done in a more descriptive or in a more practical way. For example, a course devoted to blockchain (decentralised ledger technology) could include exercises in programming smart contracts, but such a course could also be limited to helping students understand the technology at a higher level of abstraction.<sup>46</sup> Both ways may be suitable, depending on the backgrounds and aspirations of the students on the one hand and resources and knowledge of the instructors on the other.

The Cybersecurity course at Yale Law School is an excellent example of the fourth model.<sup>47</sup> It was designed by a law professor (Scott Shapiro) together with a cybersecurity expert (Sean O'Brien). The course content is not law specific and could potentially be offered to students of any discipline (perhaps even as an introductory course for computer science undergraduates). What is distinctive of the teaching method adopted in the Yale course is that it is centred on practical exercises, which include attacks on computer systems. In other words, the course aims to teach cybersecurity (defence and forensics) by teaching how to break into computer systems but does not require any previous knowledge of programming. Given the highly technical content of the course, students who do not already have a knowledge of computer programming, networking and system administration likely have to devote significant amounts of time for independent study to follow the course successfully. What makes the course challenging is also what makes it especially valuable. Completing the course equips students with the level of knowledge and skills in cybersecurity that is very rare among those who are not cybersecurity specialists and should be very helpful in a wide range of career paths. However, this level of expertise is arguably greater than what most law graduates are likely to really benefit from – or at least the course might not offer the best cost-benefit ratio for many students, which may suggest that there is space for a less technical cybersecurity course to be offered alongside this one.

The biggest concern of choosing the fourth model of teaching about technology is that it is possible that the students will not benefit as much from learning in detail about the particular aspect of computer technology as they would from gaining a broader perspective offered by courses that follow the second or even the third mod-

el. Hence, it is highly advisable to choose a topic of likely relevance to the students' future careers. From that perspective, a course on blockchain could be seen as too niche to be the only technology elective offered to undergraduate law students, but it may fit very well as one of several courses offered as part of a law and technology pathway or postgraduate degree.<sup>48</sup> On the other hand, cybersecurity is undoubtedly an issue of paramount importance to virtually all career pathways a law graduate may want to pursue. The key question regarding teaching cybersecurity concerns method: the very in-depth technical approach taken by the Yale course may be not only hard for some education providers to implement, but also unsuitable for all students (owing to differences in preparation, predispositions and career aspirations). One potential solution to that is to offer a choice of 'basic cybersecurity' (i.e. the first teaching model discussed in Section 2.1) and 'advanced cybersecurity' as different pathways within a module or separate module (or non-module teaching method like the cybersecurity fire drills I suggested in Section 2.1). As with the third model (teaching programming), the fourth model requires teachers who are technology experts. The available solutions are the same: from teachers who are interdisciplinary experts (in law and technology) to interdisciplinary teaching teams to non-law specific courses delivered solely by technology experts and potentially offered to non-law students together with law students.

## 3 How Much Should Lawyers Learn About Technology?

The question of how much technology current and aspiring lawyers should learn is being increasingly debated in academia,<sup>49</sup> on industry blogs and during industry conferences.<sup>50</sup> I suggested earlier in this article (Section 2.1) that, within the profession, it would be very

46. See D.M. Katz and N. Rosario, 'Blockchain Law Class', available at: [www.blockchainlawclass.com/](http://www.blockchainlawclass.com/), archived at: <https://perma.cc/XA3L-7B9Z> (last visited 30 January 2021).

47. Yale Law School, 'Cybersecurity', available at: <https://courses.law.yale.edu/courses/course/2793>, archived at: <https://perma.cc/27RP-Z3AP>, 'Materials for Cybersecurity (LAW 20310) at Yale Law School', available at: <https://github.com/seandiggity/yly-cybersec>, archived at: <https://perma.cc/PEZ9-8R8W> (last visited 30 January 2021).

48. This, for example, is the approach taken by the Illinois Tech – Chicago-Kent College of Law, available at: [www.thelawlab.com/courses](http://www.thelawlab.com/courses), archived at: <https://perma.cc/J3QG-SEEG> (last visited 30 January 2021).

49. See, e.g., Ashley, above n. 31; Katz, above n. 7; Williams, Janeček & Keep, above n. 7; Janeček, Williams & Keep, above n. 7; Smith and Spencer, above n. 7.

50. See, e.g., J. Krause, 'Does Learning to Code Make You a Better Lawyer?', *ABA Journal*, 1 September 2016, available at: [www.abajournal.com/magazine/article/lawyer\\_learning\\_code\\_zvenyach\\_ohm/](http://www.abajournal.com/magazine/article/lawyer_learning_code_zvenyach_ohm/), archived at: <https://perma.cc/KTK6-6HBB> (last visited 30 January 2021); L. Cheek, 'Lawyers Who Code', available at: [www.legalgeek.co/tag/code/](http://www.legalgeek.co/tag/code/), archived at: <https://perma.cc/W2FU-5QRA> (last visited 30 January 2021); B. Inkster, 'Lawyers and Coding', *The Time Blawg*, 24 February 2018, available at: <http://thetimeblawg.com/2018/02/24/lawyers-and-coding/>, archived at: <https://perma.cc/5R8C-XEQQ> (last visited 30 January 2021); Lawtomed, 'To Code or Not to Code: Should Lawyers Learn to Code?', available at: <https://lawtomed.com/to-code-or-not-to-code-should-lawyers-learn-to-code-3/>, archived at: <https://perma.cc/Q59Y-5QPM> (last visited 30 January 2021); R. Tromans, 'Should Lawyers Learn to Code? If You Have a Good Use Case, Yes', *Artificial Lawyer*, 30 January 2021, available at: [www.artificiallawyer.com/2019/06/14/should-](http://www.artificiallawyer.com/2019/06/14/should-)

valuable to spread technological literacy at a relatively basic but still higher level than likely possessed by the vast majority of current lawyers. This should include effective uses of office and legal research software and standard means of online communication and cybersecurity. As I also noted, this may be paired with teaching numerical and information literacy, on the one hand, and training in business skills, service and product design and process management, on the other.<sup>51</sup> I consider this general proposition to be relatively uncontroversial.

The more difficult question pertains to training in more advanced aspects of computer technology, including computer programming or practical data analytics. It will thus be worthwhile to summarise the key reasons for and against it. Beginning with the latter, software engineering is complex and requires significant knowledge and skills to execute at a level needed to deliver software that is being used by consumers. This concern is valid but does not mean that non-engineers should not learn computer programming. The correct lesson to draw from it is that students from other disciplines (like law) face a choice. They could be among the few who are willing to invest very considerable efforts into becoming, so to speak, fully bilingual (on a par with specialist computer programmers). I emphasise that the effort required to achieve that level of competence is, for most, incompatible with full-time practice of law (or even full-time law studies). The alternative is to learn some computer programming up to a level providing a decent measure of understanding of behind-the-scenes workings of computer technology, which may help in working together with experts on legal technology projects or simply in advising clients on technology-related legal issues. However, what is clearly a misconception is that lawyers and law students are able to take a relatively short course and that this will enable them to become fully fledged producers of legal technology.

As in the case with basic programming skills, the concern about learning effort required is not a sufficient argument against learning basic data analytics skills. As illustrated by the Pittsburgh and Iowa courses, discussed in the previous section, it is possible to learn in a semester how to extract and analyse some useful information from legal texts in a programmatic way. It may be enough to perform some research tasks for academic or legal practice purposes. This, however, leads to the second main reason for caution, which is the question of opportunity to use the skills. It is likely that not many lawyers or law graduates will have opportunities to conduct legal research in a programmatic way. For most academic and legal practice purposes, the ready-made tools from the main legal information providers (like WestLaw, LexisNexis) are sufficient. Those tools are gaining new functionalities supporting legal research, making the need for self-programmed solutions obsolete

lawyers-learn-to-code-if-you-have-a-good-use-case-yes/, archived at: <https://perma.cc/43AV-9XJT> (last visited 30 January 2021).

51. Janeček, Williams & Keep, above n. 7; Smith and Spencer, above n. 7.

in some circumstances. And even if one has a research question for which the self-programmed way would be more appropriate, they may face the problem of access to legal data (like texts of court judgments, hearing transcripts), which in some countries, like the United Kingdom, are not publicly available for machine processing.<sup>52</sup>

Turning to reasons in favour of learning advanced technology skills, I emphasise that *some* lawyers and law graduates do need them. A relatively small proportion of law graduates will be able to work as ‘legal technologists’, ‘legal engineers’ or ‘quantitative legal analysts’, developing software solutions for legal practice (programming themselves or working in interdisciplinary teams with programmers) or performing advanced legal analytics research (also in academia and in government).<sup>53</sup> Moreover, some kinds of practice of law do benefit from an intimate understanding of technology. However, just as practising construction law may benefit from vastly different non-legal expertise than practising law of patents for chemicals, it may be advisable for relevant computer technology to be taught in a more specialised way depending on the field of law, perhaps as an element of advanced optional law courses in those fields. ‘Blockchain Law Class’, developed by Katz and Rosario, which includes instruction both in technology and in relevant law, may serve as an example.<sup>54</sup>

Direct need for advanced technology skills, likely applicable to a minority of lawyers, is clearly the strongest argument for teaching such skills. There are also many other potential reasons to teach advanced technology, which by themselves are not strong enough to justify both the cost to law schools (law faculties) of providing such teaching and the significant opportunity cost to students. The opportunity cost is significant because to truly gain advanced skills, such as those taught, e.g. in Yale’s Cybersecurity course, while starting from the average level of technological skill, requires at least as much, if not more, effort as mastering a core law subject. However, those reasons are worth considering, especially that in some measure they also count in favour of basic technology education of the sort discussed earlier in this section, where both provision costs and opportunity costs are lower.

One of those weaker reasons is that students may transfer skills from some aspects of computer technology, like computer programming, to legal research and writ-

52. For instance, British and Irish Legal Information Institute, which operates the [www.bailii.org/](http://www.bailii.org/) website and publishes UK court judgments expressly prohibits ‘bulk downloading of documents’ from their website; see [www.bailii.org/bailii/copyright.html](http://www.bailii.org/bailii/copyright.html), archived at: <https://perma.cc/S66Z-JBJP> (last visited 31 January 2021).

53. Legal technology investment is growing and reached over 1 billion US dollars in 2018; N. Dolm, ‘713% Growth: Legal Tech Set an Investment Record in 2018’, *Forbes*, 15 January 2019, available at: [www.forbes.com/sites/valentinpivovarov/2019/01/15/legaltechinvestment2018/](http://www.forbes.com/sites/valentinpivovarov/2019/01/15/legaltechinvestment2018/), archived at: <https://perma.cc/SF7F-DQF7> (last visited 31 January 2021). See also Katz, above n. 7.

54. See Katz and Rosario, above n. 46.



ing.<sup>55</sup> For example, computer programming requires rigorous attention to detail, precision in writing and clarity in structuring documents. As Koch noted, both legal writing and computer programming are instances of ‘rules-driven writing’.<sup>56</sup> In computer programming, the author usually receives immediate feedback on whether they are complying with the rules of the programming language, which may help instil a habit of meticulous attention to the applicable rules while writing. The acquisition of such transferable skills may be a welcome side effect of otherwise valuable teaching, but it can hardly justify, e.g., teaching of computer programming, where students could devote the same time to the direct study of legal research and writing.

Finally, advanced study of some specific aspects of computer technology – such as learning computer programming – involves learning about many related behind-the-scenes aspects of computer technology.<sup>57</sup> Even though current and future law students may be well-versed as consumers of technology, they rarely have enough understanding of how it works, for example, to make informed decisions regarding how their use of technology affects their privacy, which is a crucial issue given the requirements of client-lawyer confidentiality.<sup>58</sup> This is not a strong reason for learning advanced skills, because there are less costly ways of bringing about the benefit of more general awareness of how technology works (e.g. basic technology education).

## 4 Conclusions: What Should Legal Education Providers Do?

Both students (including students of continuing professional education) and legal education providers should reflect on what kind of computer technology education suits their particular circumstances. There may be a worry that education providers who decide to teach, for instance, computer programming to law students are merely ‘bandwagon-jumping’, without serious and systematic consideration of the benefits that it may bring.<sup>59</sup> In this article, I have emphasised that teaching computer technology can be done in various ways and at various levels of depth and that those different ways and levels

have different cost and benefit considerations. I suggested that there are strong reasons for all current and future lawyers to acquire proficiency in effective uses of office and legal research software and standard means of online communication, basic cybersecurity and at fundamentals of quantitative thinking and methods. I also argued that advanced technology topics, like computer programming, should be taught only to the extent that this is justified by the direct need for such skills and knowledge in students’ future careers, which I predict to be true for only a minority of current lawyers and law students.

My discussion suggests a number of questions for further study. What are the outcomes of each of the teaching models discussed? Are graduates satisfied with that particular aspect of their education once they have some experience on the labour market? Does the teaching contribute to higher salaries or more satisfactory employment? Are lawyers who learned programming or data analytics any better at some typical legal tasks than others (the questions of transferability of skills)? It would be valuable to observe whether answers to those questions change over time.

What, then, should law schools (law faculties) do? On the one hand, investing in ‘teaching to code’ may be a successful marketing strategy as long as it remains a way by which law schools can differentiate themselves (i.e. if only some law schools offer it). Also, some law schools may be able to reduce the cost of providing computer technology education by benefiting from the expertise of computer science and engineering faculties within their institutions, e.g., by offering non-law specific technology education to law students (delivered by technology experts), without needing to develop law-specific modules. On the other hand, the question of how many law graduates will really benefit from more advanced technology training should be treated seriously. It may be worthwhile for some (a minority of) law schools to specialise in providing such advanced training. However, since relatively few jobs will benefit sufficiently from it and since those interested in learning advanced technology topics have access to a plethora of excellent online learning options (including free ones), most law schools should think twice about taking this route. What all law schools should do in terms of technology education is either provide training in what I referred to as the basic technological and numerical literacy or at least actively encourage students to learn them from some of the excellent internet resources available.

55. K.L. Koch, ‘A Multidisciplinary Comparison of Rules-Driven Writing: Similarities in Legal Writing, Biology Research Articles, and Computer Programming’, 55 *Journal of Legal Education* 234 (2005).

56. *Ibid.*, at 237.

57. M. Fenwick, W.A. Kaal & E.P.M. Vermeulen, ‘Legal Education in the Blockchain Revolution’, 20 *Vanderbilt Journal of Entertainment and Technology Law* 351, at 382 (2017).

58. See e.g. Tirumala, Sarrafzadeh & Pang, above n. 19; Thompson and others, above n. 19.

59. See e.g. A. Young-Powell, ‘More Universities are Teaching Lawtech – But Is It Just a Gimmick?’, *The Guardian*, 12 April 2019, available at: [www.theguardian.com/law/2019/apr/12/more-universities-are-teaching-lawtech-but-is-it-just-a-gimmick](http://www.theguardian.com/law/2019/apr/12/more-universities-are-teaching-lawtech-but-is-it-just-a-gimmick), archived at: <https://perma.cc/QJC6-DP26> (last visited 31 January 2021).