The AI-Powered Workplace

How Artificial Intelligence, Data, and Messaging Platforms Are Defining the Future of Work

Ronald Ashri

Apress®

THE AI-POWERED WORKPLACE

HOW ARTIFICIAL INTELLIGENCE, DATA, AND MESSAGING PLATFORMS ARE DEFINING THE FUTURE OF WORK

Ronald Ashri

Apress®

The AI-Powered Workplace: How Artificial Intelligence, Data, and Messaging Platforms Are Defining the Future of Work

Ronald Ashri
Ragusa, Italy

ISBN-13 (pbk): 978-1-4842-5475-2 ISBN-13 (electronic): 978-1-4842-5476-9
https://doi.org/10.1007/978-1-4842-5476-9

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Shiva Ramachandran
Development Editor: Laura Berendson
Coordinating Editor: Rita Fernando

Cover designed by eStudioCalamar

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail rights@apress.com, or visit www.apress.com/rights-permissions.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at www.apress.com/bulk-sales.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at www.apress.com/9781484254752. For more detailed information, please visit www.apress.com/source-code.

Printed on acid-free paper

To Katia, Sofia, and Leo.

Contents

About the Author

Ronald Ashri is always trying to find a balance between his appreciation of academic rigor and his attraction to the necessary chaos of creating practical, usable tools. This split also describes his working life, from PhD student to research fellow to a technically focused entrepreneur and consultant. For the past 15 years he has been either building products in startups or working for organizations such as BT Labs, the NHS, TripAdvisor, the Italian Government, the UK government, UCLA, McGill University, and BDO to help them build useful products. Most recently he is the cofounder of a conversational AI consultancy called GreenShoot Labs, based in London, and leading the development of an open source conversational application management platform called OpenDialog.ai.

Ronald specializes in AI systems design, knowledge management, agent-based systems, and conversational AI. He frequently writes and speaks about AI-related issues, has authored and coauthored a variety of articles (both academic and not) on AI as well as wider software engineering issues, and is the coauthor of a book titled *Agent-Based Software Development* (Artech House Publishers, 2004).

He holds a BSc (First Class) in Computer Systems Engineering from Warwick University and a PhD in Computer Science from Southampton University.

Acknowledgments

Huge thanks to Shiva for getting in touch with the idea of writing a book. Quite simply this would not have happened if she did not kick-start the process. Rita Fernando and Shivangi Ramachandran from Apress have been a huge help throughout the entire process, gently nudging along and encouraging where required or equally gently but more directly telling me to "get it done" where that was required! Thanks to both of you for your patience and enthusiasm throughout.

I got involved with AI because of Prof. Michael Luck. A little over 20 years ago I walked into his office at Warwick University to meet the lecturer who would be my tutor. From that first day, through supervision of my PhD degree, to research work afterward, and then on to a long friendship, he has always given me invaluable advice, support, and help.

Tim Deeson, my cofounder at GreenShoot Labs, is the person to whom a lot of the ideas in this book should be attributed. It is through countless discussions with him about conversational applications and AI and the challenges of introducing new ideas in organizations that the concepts got their current shape and form. Thanks for the trust he showed in me from the start.

Ultimately, however, none of this would have been possible for me without the unquestioning support, enthusiasm, and love of my wife Katia and the love and joy that our children, Sofia and Leo, bring to us daily. Thanks to the three of you for your patience while I locked myself away and wrote, and your support as I went through the ups and downs of this process.

Introduction

What does the phrase "The AI-powered workplace" mean? Is it a blatant attempt to ride a wave of interest in artificial intelligence (AI) or does it describe a set of tools, techniques, and methods that can provide real value and lead to a different kind of workplace? In a broader sense, are we at the start of a fundamental shift in how office work is done, with AI as one of the central pillars, or are we in the throes of the latest fad?

Unsurprisingly, I don't think that AI is a fad. I believe that we are at the start of what will be a far-reaching and fast-paced journey, one that will lead us to a radically different workplace from what we've had for most of the second half of the 20th century and a good chunk of the 21st.

The tools to help us get there are broadly, and confusingly, referred to as AI technologies. These technologies will prove to be a fundamental game-changer in how we do work. Indeed, AI will be a significant game-changer in all aspects of life, from manufacturing to agriculture and from healthcare to, sadly, also warfare.

Our focus for this book, however, will stick to the office. We will explore what changes are already underway and which are next on the horizon. These changes will affect the type of work we do when we are staring at a computer screen (and how we will hopefully be doing much less staring as time goes by) and coordinating with our coworkers to get things done. More specifically, we will explore how the rise of messaging and collaboration platforms such as Slack, Microsoft Teams, or Facebook Workplace provide fertile ground for the growth of increasingly more sophisticated applications that work with us in a proactive manner to help us solve problems. These messaging platforms or, as we will more appropriately term them later on, conversational collaboration platforms, will turn into the operating system of our digital workplace. Understanding how AI and these platforms intersect can provide you with a jump-start toward an AI-powered workplace.

The goal of this book is very practical: to equip you with a solid understanding of what AI is and provide you with decision-making tools that will allow you to chart a plan of action for your team that takes into account and exploits what AI technologies coupled with messaging platforms can offer.

The book is divided into broadly three sections. First, we provide an introduction to AI and a way to think of AI-powered applications that will stand us in good stead now and in the future. We then turn our focus to the application of AI in the workplace, with a specific emphasis on how conversational collaboration platforms motivate and facilitate the introduction of AI-powered applications. Finally, we deal with the strategic thinking that is required to help you build your own AI-powered workplace by looking at useful principles and methods to develop strategies around data and AI in general. We close with a look at the ethical issues that are raised by the use of AI and discuss some of the guidelines that are being developed to tackle those.

Understanding AI

In Chapter 1 we trace the rise of AI and its trusty friends Data and Processing Power. This will help us better appreciate why we are where we are now, what the critical phases were in getting here, and how those will influence future developments.

In Chapter 2 we tackle the big question. What is AI and how does it do what it does? Don't worry; I am not about to put on a philosopher's hat and ponder the meaning of intelligence. My goal is to provide a set of practical ways of thinking about AI, demystifying the technologies and allowing you to recognize what can be practically useful in your situation. In particular, I will describe a way of thinking about *what* AI-powered applications do that is separate and distinct from the techniques that they use to achieve it.

Equipped with an understanding of what AI is, in Chapter 3, we look at what it means to build an AI-powered application. We look at what it takes to come up with a model of reasoning that will empower automated decision-making. In particular, we explore two distinct approaches: model-driven AI techniques and data-driven techniques. Both are required for complete applications, so it is useful to have a clear understanding of the implications, benefits, and disadvantages of each.

Chapters 4 and 5 build on the framework introduced in Chapter 3. They are the most "technical" chapters, in that they provide us with an overview of the range of AI techniques and capabilities currently available and widely used. The purpose of these chapters is to give us a handle on various terms we will inevitably encounter, such as supervised or unsupervised learning and natural language processing, so that they can be considered in the context of solutions.

Applications of AI in the Workplace

In Chapter 6 we take a brief look at what we mean by the "digital workplace." This helps define the space we will be looking at when thinking of applications of AI in the workplace and strategies of how to transform that workplace.

In Chapter 7 we consider interface paradigms. The way we interact with digital machines is changing as digital machines are able to support richer and more natural interactions. Whether we are dealing with augmented reality; virtual reality; voice; or plainly but perhaps most effectively, text AI, plays a key role. Understanding how the rise of AI and the emergence of a new set of paradigms are interlinked is crucial in helping you understand how it may impact your workplace and why conversational collaboration platforms accelerate the adoption of AI.

In Chapters 8 and 9 we take a deep dive into conversational collaboration platforms, their components, the considerations that come into play, and what it means to build chatbots, or better still, conversational applications. We examine how applications such as Slack, Microsoft Teams, and Facebook Workplace are not only changing how teams work, but also how automation and intelligent services can be introduced in the workplace. We outline a few different ways of thinking about conversational applications and provide an overall conceptual framework that ties it all together.

Building Your AI-Powered Workplace

The goal of the last three chapters is to provide us with some practical and directly actionable ways to approach the strategic development of our AI-powered workplace.

In Chapter 10 we discuss data, what role it plays, how various challenges around it can be approached, and through a practical example give some directions of how you can develop your own data strategy. We eschew consultancy-speak, with lofty visions and fancy terms focusing instead on pragmatic steps you can take.

In a similar fashion, Chapter 11 looks at useful elements to consider when devising a wider strategy for AI applications. Once more, the objective is to provide actionable insights that recognize that each organization needs to develop their own strategy, specific to their own context. As such we look at overarching principles and a set of methods that I have seen work successfully in the past.

Finally, in Chapter 12 we look at the ethical considerations of building AI applications. We start by explaining why these issues need to be considered specifically and explicitly within the context of AI-powered applications and then consider overall guidelines to get us started on the process. These are crucial issues, and it is down to our individual and collective responsibility to ensure that we critically consider them as we introduce increasing levels of automation and autonomy in our teams.

In all, I hope this is going to be an interesting journey for everyone. If you have never dealt with AI subjects before, this should give you a solid primer and a way to start taking practical actions while researching further. If you have some experience, this book could help round out some specific subjects and give you a couple of different perspectives on how to consider AI tools in the context of conversational platforms, as well as how to go about defining an approach that you can apply in your own organization.

PART

I

Understanding AI

CHAPTER

1

The Search for Thinking Machines

The wish to construct machines that can perform tasks just like us or better than us is as old as our ability to reason about the world and question how things work.

In the *Iliad*, when Thetis goes to ask Hephaestus for replacement armor for her son Achilles, Homer describes Hephaestus's lab as a veritable den of robotics. There are machines on tripods whose task is to attend meetings of the gods (yes, even the gods hated going to meetings themselves), robotic voice-controlled assembly lines, and robots made out of gold to help their master.

> *They were made of gold but looked like real girls and could not only speak and use their limbs but were also endowed with intelligence and had learned their skills from the immortal gods. While they scurried around to support their lord, Hephaestus moved unsteadily to where Thetis was seated.*[1]

[1] *Iliad* 18, 418-422.

R. Ashri, *The AI-Powered Workplace*,
https://doi.org/10.1007/978-1-4842-5476-9_1

Building robots was the job of gods. They breathed life into machines. Homer was telling us that the ability to create thinking machines could bestow on us god-like status.

The challenge we had back then, and still have, is understanding exactly how we might go about building such machines. While we could imagine their existence, we didn't have the tools or methods that would allow us to chart a path to an actual machine. It is no wonder that very often when solutions were imagined, they included secret potions and alchemy that would magically breathe life into Frankenstein-like figures not entirely under our control.

Step by step, though, we have put some of the pieces of the puzzle together. At the mechanical level, humans managed to build very convincing automatons. Through clever tricks, the creators of these automatons even fooled people into believing they were magically endowed with intelligent thought. From the ancient Greeks to the Han dynasty in China, the water-operated automatons of al-Jazarī in Mesopotamia, and Leonardo da Vinci's knight, we have always tried to figure out how to get mechanical objects to move like real-life objects while tricking the observer into thinking there is an intelligent force within them causing action.

From a reasoning perspective, we went from the beginnings of formal reasoning, such as Aristotle's syllogisms, to understanding how we can describe complex processes through algorithms (the work of al-Khwārizmī around 820 A.D.), through to Boole's *The Laws of Thought*, which gave us formal mathematical rigor.

Nevertheless, we were still woefully ill-equipped to create complex systems. While lay people could be fooled through smoke and mirrors, the practitioners of the field knew that their systems where nowhere near close to the level of complexity of human (or any form of natural) intelligence.

As advances marched on and we got to the 19th century, the field of computer science started taking shape. With Charles Babbage's Analytical Engine and Ada Lovelace's work on programming, the outlines of a path from imagination to realization started emerging. Ada Lovelace speculated that the Analytical Engine, with her programming, "might compose elaborate and scientific pieces of music of any degree of complexity or extent."

By the end of the Second World War, we had progress in computing that was the result of efforts to build large code-breaking machines for the war, and the theoretical advances made by Alan Turing. A few years later (1950), Alan Turing even provided us with a test to apply to machines that claim to be able to think. The test calls for a human being to hold a conversation with a machine while a third observer is able to follow the conversation (by seeing what was said printed out). If the third observer cannot distinguish between the human and the machine, the machine passes the test.[2]

The path toward artificial intelligence was getting increasingly clearer.

The Birth of a New Field of Study

On August 31, 1955 a group of researchers in the United States produced a brief document[3] asking for funding for a summer research project. The opening paragraph is a testament to the unbounded optimism of humans and a lesson in what the phrase "hindsight is everything" means.

Here it is:

> *We propose that a 2 month, 10 man study of artificial intelligence be carried out during the summer of 1956 at Dartmouth College in Hanover, New Hampshire. The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it. An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves. We think that a significant advance can be made in one or more of these problems if a carefully selected group of scientists work on it together for a summer.*

The most striking phrase, to a 21st century reader, is the very last one. It claims that a significant advance can be made after just a single summer of work. Keep in mind that this was with 1956 computing technology. The first integrated transistor was 4 years away. Computers occupied entire rooms and logic modules used vacuum tubes.

Now, to be fair, the group of scientists in question was exceptional. The proposal was co-signed by four very influential computer scientists. Claude Shannon, the father of information theory; Marvin Minsky, one of the first to build a randomly wired neural network; Nathaniel Rochester, the lead designer of the IBM 701, the first general-purpose, mass produced computer; and John McCarthy, widely credited with coining the term "artificial intelligence" and the creator of the Lisp programming language. If any group of people stood a chance of making a significant breakthrough in 1956, this was certainly it.

This meeting in Dartmouth is generally credited as the first conference on AI. It shaped academic thought around the range of issues to be dealt with in

---

2 *Alan* Turing, "Computing Machinery and Intelligence" *in Mind,* volume LIX (236) *(*Oxford University Press, 1950) pp. 433–460.
3 www-formal.stanford.edu/jmc/history/dartmouth/dartmouth.html

order to create machines that exhibit intelligence. It also produced a lot of excitement and *buzz* and led to several years of funding for AI, largely based on the trust people had in people like Minsky to pull the proverbial rabbit out of the hat.

Those *golden* (in an almost literal sense) years laid the foundations of the field of AI. From 1955 to the early 1970s, a whole range of subfields was created, and problems and their challenges stated. From different ways to reason to a start at natural language understanding, the first neural networks, and so much more, the field was booming.

As ever, however, buzz and excitement, unaccompanied by the results that people expected or predicted, led to what was known as the first AI winter. People, initially bedazzled by hyperbolic claims, were eventually disillusioned with AI and lost hope in its ability to produce results. AI was sidelined, funding was reduced, and focus turned to other issues.

It's important to note that research in AI did not stop over this period. It may have been less visible to the public eye, and overall resources were reduced, but interest remained. Research grants used the term "AI" less often, but they were still trying to solve the same problems.

A Practical Application

In the 1980s there was a resurgence. AI had found a problem that it could solve for businesses in a manner that was widely applicable and where returns were clear. By that time, several research groups, but in particular the Stanford Heuristic Programming Project, came to the realization that instead of trying to create general problem solvers, they could instead focus on constrained domains that required expert knowledge. These *expert systems* used all the research that happened in the previous decades about how to codify knowledge and reason about it, and focused it on specific use cases within restricted domains.

Expert systems can broadly be described as the combination of codifying knowledge as a set of rules and creating an inference engine that can take in a description of the state of things and derive some conclusions. Roughly, you would have a collection of rules such as "If temperature is below 60°F, wear a warm coat" and relationships such as "A jacket is a type of coat." Combining large numbers of such rules and relationships, expert systems can capture an expert's highly complicated knowledge in a specific field and act as an aid to augment human-based expertise. By the mid-1980s there was a whole industry of companies focused on supplying technology to run expert systems. These software behemoths used programming languages such as Lisp to encapsulated knowledge, and used dedicated machines to be able to crunch through the rules.

The iconic expert system example from the 1980s was called XCON (the eXpert CONfigurer), built for Digital Equipment Corporation. XCON's job was to assist in the ordering of DEC computers. The inputs were the client's requirements and the outputs were the necessary computer system components. XCON was built based on the knowledge of DEC's experts and had around 2,500 rules. It ended up saving up to 25 million dollars a year because it processed orders with a higher level of accuracy (between 95% and 98%) than technicians, thereby reducing the number of free components DEC had to send out following a mistaken configuration.

With expert systems, entrepreneurs saw an opportunity; AI had found its killer feature, and business was booming again. Once more though, hype got the better of everyone. Big claims were made and, while a lot of useful software was built and was effectively being used, the expectations were set too high. As a result, that industry crashed and several companies disappeared. AI was in the doghouse once more.

What is a less-often mentioned feature of this second AI winter is that a lot of the industry (i.e., a lot of the money involved) was focused on building specialized machines able to run expert systems. It's important to keep in mind that the PC industry was still confined to hobbyists at this time. What the expert systems industry did not foresee was the rise of PCs. They became popular and were recognized as valid and useful business machines, making the expensive expert systems machines seem less attractive.

Quite simply, there was no need for a large industry building actual computers to run expert systems, because people could get a lot done with normal and cheaper PCs. Business could do digital transformation in a much more agile way by introducing PCs.[4] Giving everyone a word processor and a flexible number cruncher and letting them figure out how to make good use of them seemed like a much better way to invest money at the time. Part of the story, therefore, is that the technological landscape changed, and the AI industry failed to adapt quickly enough.

Businesses use technology to gain a competitive advantage. At the time, investing in large and complicated AI systems was risky. It made much more sense to bring the entire company up to date with computing technology and empower many more people within the company to benefit from general computing improvements.

Understanding these ebbs and flows is fundamental in learning to distinguish noise from signal. All through those 40 years (from the early 1960s to the early 2000s) capable people built capable software that solved real problems using AI. They also advanced the science of how we could solve complex

[4] While they may not have used the terminology "digital transformation" and "agile" in the 1980s, that is exactly what they were doing by putting PCs in front of everyone.

problems with computers. The battle was not really fought or measured in that respect. The AI winters or springs were measured in the ability of people to attract funding (which induces people into making lofty claims) and the competition AI had from other types of technologies vying for investment. By the early 1990s AI generally had a tarnished name, and there were so many new, exciting, and much more immediately applicable technologies to invest in (like a little something called the World Wide Web).

AI Goes into Hiding

As a result of all of this, AI researchers devised a different survival strategy. Instead of saying they were working on AI, they would focus on talking about the specific subfield they worked in. A whole range of different fields emerged, from multiagent systems (the field I worked on as a PhD student and researcher) to knowledge management, business rules management, machine learning, planning, and much more. What we were all doing was trying to build machines that solve problems in a better way. We just stopped emphasizing the AI part that much. Even the researchers who were directly focused on the hardest task of them all, using computers to understand or recreate human behavior, labeled it as "cognitive computing" or "cognitive sciences" rather than calling it research in artificial intelligence.

In the background, gatherings such as the International Joint Conference on Artificial Intelligence (running every 2 years from 1969 to 2015 and now, incidentally, running yearly) remained. The people who were reticent to talk about AI on funding proposals still attended. Everyone knew they were working on AI, but they would not necessarily hype it as much.

My own personal path to AI reminds me of just how much AI was out of favor. As an undergraduate at Warwick University, in 1999, I had to get "special" permission from the Computer Science department to attend a cognitive computing course because it was taught in the Psychology department! I ended up loading my undergrad studies with so many AI-related courses that I was called in to justify it as still a valid degree in Computer Science. Luckily, all the AI academics-in-hiding had my back and the argument was won. Fast-forward to 2019 and universities are rushing to introduce Artificial Intelligence degrees while academics are reclaiming titles such as Professor of Artificial Intelligence, when only 5 years ago they were "plain" Professors of Computer Science.

What all this helps illustrate and what is essential to keep in mind when you hear about AI having gone through cycles before and AI being just hype, is that AI is an extremely broad term. It is such a broad term that it is arguably close to useless—unless, that is, it captures the imagination of investors and the popular press. The problems it is trying to solve, however, can't go away. They are real scientific questions or real challenges that humanity has to tackle.

Separating the noise of the press and the hype that gets investors excited from what matters is crucial in approaching AI. In the next section, I will explain why I think we have reached a point where talking about cycles of AI is not very useful, and in the next chapter we will get a better handle on what AI actually means.

The New Eternal Spring

Since the late 2000s to now, 2019, excitement about AI feels like it has reached fever pitch and will need to settle down a bit. At the same time, a lot of other things have been happening in the world of technology and outside of it. These other elements have fundamentally changed the landscape once more.

This time around, unlike the 1980s, I believe that the way the landscape is changing is in favor of the long-term growth of AI and the growing industry of AI-related technologies. The changes are not only going to help spread AI, they are going to make AI a necessity. AI will stop being a novelty application that is occasionally introduced. Instead, it will be one of the fundamental pillars woven into the fabric of everything that we do.

We may call it different names and investors may lose interest once more, but the tools and the solutions to problems they provide will remain. We can already see this happening explicitly with smartphones, where dedicated chips for AI calculations are introduced. If those were switched off, we wouldn't be able to even open the phones (face recognition) or type on them (predictive typing).

At the most basic level, there are at least three forces in play here that are going to necessitate the adoption of AI techniques:

1. We have an unprecedented increase in the amount of data we handle, and the need to carefully curate it and base decisions on it.
2. Processing power, which has followed Moore's law for the past 50 years, is making it possible to apply complex algorithms to data. The performance of these algorithms for well-defined domains is equaling or surpassing human abilities. At the very least, these make the tools useful companions, *augmenting human capabilities*.
3. Cloud computing is making both storage and processing power widely available with an added layer of sophistication, so that anyone can access and make use of complex AI tools. Open source tooling is doing the same in terms of the lower level capabilities. AI is being democratized in a way that was never possible before.

Flooded in Data

The first fundamental shift is the extent to which what we can do with data has changed. Data has always been produced. The question is whether it could be captured, stored, retrieved, and analyzed. In that respect, the past years have introduced magnitudes of change. We can now, very cost effectively, store huge amounts of data (although most of it is still unstructured data), and we can process huge chunks of the data that we do store.

It is hard to say exactly how much data we store daily. Here are some numbers presented by IBM in 2018:

- 80,000,000 MRIs taken every year
- 600,000,000 forms of malware introduced daily
- 2,200,000,000 locations generating hyperlocal weather forecasts every 15 minutes

If you head over to internetlivestats.com, you will see a dizzying number of counters ticking up. When I checked it one mid-morning in September 2019 from Central Europe, the numbers that stood out were:

- Over 115,000,000,000 emails sent in the day already
- Over 35,000,000 photos uploaded to Instagram
- Over 3,000,000,000 searches on Google already

For once it is no exaggeration to use terms like "flooded" or "inundated" to describe the situation with data. According to research done by *Raconteur*, we are likely to reach 463 exabytes[5] of data produced daily by 2025.[6] That is going to be driven to a large extent by the increase in devices that are connected to the Internet of Things (IoT) in wearables, smart devices around the home, and in industry.

The AI opportunity is simply that humans can in no way, shape, or form expect to analyze even a tiny fraction of this data without automation. Applying large-scale data analysis to it is our only hope. Taking this a step further, we can also expect that unless we embed automated decision-making into systems, we will not be able to cope with the growth in demand for decision making and action.

To consider some of the staggering needs humanity has, here is one example from education. According to goals adopted by the UN General Assembly, in

[5] It is hard to capture just how much data 463 exabytes are. According to a 2003 report from Berkley University, all words ever spoken by human beings would take up about 5 exabytes. So we will be producing 93 times that data, a day, every day! (Peter Lyman and Hal R Varian, *How Much Information?*. Technical Report (Berkeley UC Berkeley, 2003).

[6] http://res.cloudinary.com/yumyoshojin/image/upload/v1/pdf/future-data-2019.pdf.

order to meet a 2030 education goal of providing every child with primary and secondary education, an additional 69 million teachers will be needed across the world.[7] As of 2018, there were 64 million teachers around the world[8] (this is ignoring quality issues such as level of training of the teachers themselves). We would need to double the number of existing teachers worldwide in 12 years. To train enough teachers to meet our goals, we need a huge amount of support from technology to scale teacher training, scale pupil education, and more accurately measure results. All of this will depend on effectively managing data and automating processes, to allow human resources to focus on what they can do best—person-to-person interactions.

In China, a company called Squirrel AI has managed in 2 years to build over 700 schools for K–12 education, catering to over 1 million students.[9] The entire system depends on sophisticated algorithms that are able to adapt a teaching program to a student's individual needs. While it is healthy to be skeptical about how such systems will really impact children and education, we have to, at the same time, accept that the exploration of such solutions is the only way to manage a growing population and growing needs for education.

Raw Processing Power and Better Algorithms

What do first-person shooter games and AI have in common?

Believe it or not, without the former we might not have had quite the resurgence of the latter. In 2009, at the International Conference on Machine Learning, Raina et al.[10] presented a paper on the use of graphics processors applied to large-scale deep belief networks. Raina's research group at Stanford, led at the time by the now well-known Andrew Ng,[11] figured out how to effectively use graphical processing units (GPUs), specifically through NVidia's CUDA programming model, to dramatically decrease the time it took to train machine learning models by exploiting the parallelization afforded by GPUs. By that time, GPUs were so efficient *and* very cost-effective because of the

[7] www.unesco.org/new/en/media-services/single-view/news/close_to_69_million_new_teachers_needed_to_reach_2030_educat/.
[8] https://tellmaps.com/uis/teachers/#!/tellmap/1381436086.
[9] http://www.squirrelai.us/school.html.
[10] Rajatm Raina, Anand Madhavan, and Andrew Y Ng, "Large-Scale Deep Unsupervised Learning using Graphics Processors," International Conference on Machine Learning (Montreal, Canada: 2009).
[11] Andrew Ng was Director of the Artificial Intelligence Lab at Stanford, then became the director of Google Brain, cofounded Coursera, and led AI efforts at Baidu. In 2017 he left Baidu to create Deeplearning.ai (offering online AI classes) and Landing.ai, which is focusing on the use of AI in manufacturing.

increase in sales of GPUs to power the needs of the computer gaming industry for, point in question, better graphics for first-person shooter games.

The use of GPUs, coupled with advances in algorithm design (in particular from Geoffrey Hinton's group at the University of Toronto), led to a step change in the quality of results and the effort it took to arrive at those results.

In many ways, these two advances have released the genie out of the bottle. AI as an industry may suffer at some point, investors may complain, and we may not get all the things that are being promised. The techniques developed so far, however, will always be available, and because of their wide applicability they will always be used to solve problems—even more so as we see access to AI democratized through cloud computing and open source software.

The Democratization of AI

Up until 2015, actually using AI technologies was generally hard. Unless you had AI experts in your team, you couldn't realistically attempt to apply any of these technologies. A lot has changed since then.

In November 2015, TensorFlow[12] (Google's framework for building machine learning applications) was released as an open source tool. Since then, the open source space of AI tooling has exploded. Online courses abound, both paid for and free, and the main blocker to learning about AI techniques is your time availability and access to bandwidth.

In addition, by 2019 we have an unprecedented level of access to sophisticated machine learning services through simplified application programming interfaces. Amazon, Microsoft, Google, IBM, and many others offer access to their tools, allowing organizations to upload data, train models, and deploy solutions within their applications.

Microsoft, in particular, talks directly about democratizing AI.[13] The aim is to make it available to all software engineers in the same way database technologies or raw computing power is available via cloud-based solutions.

The wide availability of these technologies as cloud solutions doesn't just reduce the level of expertise required to implement an AI-powered application. It also reduces the amount of time it takes to go from idea to prototype to production-level roll-out. As we will see later on, the ability to iterate, experiment, and learn while doing is just as important as having the technologies readily available.

[12] www.tensorflow.org/.
[13] https://news.microsoft.com/features/democratizing-ai/.

Moving Forward

In the last section we laid out the case for why AI cannot simply disappear. Please note that this is not about the economics of the AI industry. It is not about the venture capitalists, startups, or large corporation and government politics of who gets more attention and funding. The argument is about the fundamentals of how the digital world is evolving and the needs of the analog world. The problems are getting bigger, and we are running out of ways to scale systems unless we introduce automation.

Whether it is agriculture, education, health, or work in the office, we cannot just keep working longer and stressing out more. Something has to give and something has to change. My hope is that what is going to change is more efficient use of technology to allow us to focus on what is more important. Yes, along the way we will go through booms and busts. However, do not confuse press coverage or funding news about unicorn startups with the fundamentals. The goal of AI should not be to make us all more like machines or to create unicorn startups.

The goal of AI, as that of any technology, should be to lift us up from our current condition and give us back the time to explore what it means to be human.

CHAPTER

2

What Is AI?

Artificial intelligence (AI) is notoriously hard to define, and this has been both a boon and a curse. The broadness of the term has allowed for a very wide and disparate set of techniques to inhabit the same space, from data-intensive machine learning techniques such as neural networks to model-based deduction logics, and from the incorporation of techniques from statistics to the use of psychological models of the mind. At the same time, all these attempts to emulate existing forms of intelligence or create new ones has allowed for debates to flourish about what is actual intelligence. While in some contexts these debates are useful, they can also be distracting and confusing. They can lead people to set expectations or express concerns about AI that are not founded in what the technology is currently capable of or what we can confidently say it will be capable of, but rather extrapolations that are based more on beliefs and hunches.

In this chapter we are going to focus on a more practical interpretation of AI. To set the scene, we start by briefly looking at one of the biggest traps of defining AI, what is often referred to as its magical "disappearing act." We then distinguish between general AI and domain-specific AI, and move on to provide a way of thinking of domain-specific AI that focusses on the *what* without concerning itself with the *how*. By focusing on outcomes and observable behavior, we can largely sidestep the problem of having to even use a term as fluid as intelligence, and certainly we will not be trying to pin it to a definition. Instead, we ground our thinking in how our organizational processes are affected by the introduction of software that has been delegated some aspect of decision-making and, by consequence, the automation of action. All of the aforementioned provides the necessary groundwork to support the next chapter around building AI-powered applications.

R. Ashri, *The AI-Powered Workplace*,
https://doi.org/10.1007/978-1-4842-5476-9_2

AI's Disappearing Act

Based on what we saw about the history of AI in the previous chapter, it is reasonable to say that "artificial" in artificial intelligence refers to the fact that the origin of "intelligence" is the result of purposeful human effort, rather than natural evolution or some form of godly intervention. It is "artificial" only because we generally thought of intelligence as something that came from nature, whereas with AI we somehow trick nature and create it ourselves.

I like to think of it as very similar to synthetic elements in the periodic table. If you recall from your chemistry classes, synthetic elements are those that have been created artificially, that is, by humans. You generally will not find Einsteinium or Fermium lying around. What is important though, both for these synthetic elements and for AI, is that once created they are no less or more real than the elements (or the intelligence) that can be found in nature. In other words, artificial refers to the process of arriving to intelligence, *not* the final result.

What is intelligence then? This is about as open-ended a question as you can get, compounded by the fact that we make things worse for ourselves by constantly changing the target. Every time we build something that can do something that wasn't possible before, from the simple calculator to beating chess grandmasters, or from winning at *Jeopardy!* to detecting cancer, that intractable problem we just made tractable gets declassified. The typical dialog with the person claiming that something is not real intelligence broadly follows these outlines:

- It *looked* like it required intelligence to be solved, but I guess it doesn't.
- Wait, what? So how is it solved now, if not by AI?
- Well it uses lots of computing and lots of algorithms, doesn't it? There is nothing really smart or magical about it.
- …

The minute the trick is revealed, the spell is broken and it is magic no more. It's sleight of hand, an elaborate trick, a more sophisticated version of a singing bird automaton.

Perhaps this is a result of the fact that we don't actually know where and how we get our own intelligence, combined with an innate sense of threat at anything claiming that it might be intelligent like us. After all, we are quite used to ruling the roost on this planet. Our own search for intelligence is the only thing that feels like it might threaten it.[1] When we manage to solve problems we "thought" required intelligence, we demystify them, and that makes them less interesting. Until such a time when we have demystified the entire process, we can always retreat to higher ground and claim superiority.

For the day-to-day task of solving real problems, however, such discussions tend to distract and discourage. It ultimately doesn't matter whether people call it intelligence or not. In the most banal of ways, it is the journey to get to the solution that matters. That journey takes us through the wide toolset that AI offers, and allows us to discover what techniques will work for our case. Whether we've solved a problem using "real" intelligence or not is a debate for existentialists. The fact on the ground is that we now have a tool that does something useful.

It is this practical view of AI that we will develop further in this chapter. Before we do so, however, we are going to distinguish between the search for general AI, which is definitely an existentialist pursuit, and domain-specific AI.

General AI vs. Domain-Specific AI

As we move toward a practical understanding of AI, so that we can use it to reason about how best to exploit it, it's useful to start by distinguishing between artificial general intelligence (or strong AI) and domain-specific (or weak AI).

Strong AI refers to the effort to create machines that are able to tackle any problem by applying their skills. Just like humans, they can examine a situation and make best use of the resources at hand to achieve their objectives.

Take a moment to consider what that means. Say the objective is to prepare a cup of coffee. Think of doing that in a stranger's house. You are let in, and then you scan the rooms trying to figure out where the kitchen might be. Years of prior experience tell you it is probably on the ground floor and toward the back of the house. At any rate, you can recognize it when you see it, right? You then look around for a coffee machine. Does it take ground coffee, beans, or soluble? Is it an Italian-style coffee maker, or a French coffee press? Where is the water? Where do they keep cups? Spoons? Sugar? Cream? Milk? We breeze through all these problems without a second thought.

---

[1] Note that I said "feels like it might threaten it." There is an intense debate about the risks AI poses to humanity from an existential point of view. That is one debate that this book will not try to tackle. However, I do urge everyone to consider the near-term risks stemming from the misuse of current day technology (a risk I consider urgent and present), as well as investigate carefully if there are any real long-term risks from sentient software taking over (risks I consider more of a thought exercise than, in any form, real).

Now imagine having to build a machine that will do this. It would require navigation skills, machine vision skills, dexterity to handle different types of objects, and a library's worth of rules on how coffee is made and how homes are structured. You are probably beginning to appreciate the challenges strong AI has.

The coffee-making scenario is a challenge that Steve Wozniak (yes, Apple's Steve Wozniak) devised as a test for a strong AI machine. Just like the Turing test we mentioned in the previous chapter, it is a way to verify whether anything close to human-level intelligence has been reached. The catch is that even if you do build a machine that is able to enter any house and prepare a cup of coffee (which, incidentally, we are not anywhere close to achieving), it will fail woefully the minute you ask it to change a light bulb. In fact, many argue that even this test is not a sufficiently good test of strong AI. A crucial skill of general intelligence is the ability to transfer knowledge from one domain to the other, something that humans seem uniquely capable of doing. This quote from the "Ethics of Artificial Intelligence" captures this nicely for me.

> *A bee exhibits competence at building hives; a beaver exhibits competence at building dams; but a bee doesn't build [a] dam, and a beaver can't learn to build a hive. A human, watching, can learn to do both; but that is a unique ability among biological lifeforms.*[2]

Strong AI is trying to tackle questions that go straight to the core of who we are as humans. Needless to say, if we ever do build machines that are so capable, we will have a very interesting further set of questions to answer.

As such, the debate is fascinating from a philosophical, political, and social perspective. From a scientific perspective the search is certainly worthwhile. From a "how can AI help me get the work I have in front of me today done?" perspective, strong AI is not where we need to be focusing.

We will, instead, focus on weak or narrow AI. This is AI that is trying to build machines that are able to solve problems in well-defined domains. Similar to the expert systems of the early 1980s with their "few" thousands of rules, the goal of these AI machines is to solve delimited problems and demonstrate their value early and clearly. What differs from the 1980s is that we now have the computing power, data, and techniques to build systems that can solve problems without us having to explicitly articulate all the rules.

[2] Nick Bostrom and Eliezer Yudkowsky, "The Ethics of Artificial Intelligence" in *The Cambridge Handbook of Artificial Intelligence* (Cambridge, UK: Cambridge University Press, 2014) pp. 316–334.

In addition, instead of diving straight into technologies and providing taxonomies of the different types of machine learning or symbolic reasoning approaches, we are going to take a different path. Since intelligence is such a hard thing to pin down, we are going to look at the qualities or behaviors a system displays and use those to understand it. We are going to draw on ideas from agent-oriented computing, a field of AI that occupies itself with building software wherein the core unit of abstraction is an agent. We will explore what sort of behaviors our agents (i.e., our software) can have, and how these behaviors combine to lead to increasingly more sophisticated software.

An Agent-Based View of AI

Perspective has the powerful capability to define how you are able to understand a problem. Approaching AI from an agent-oriented view liberates us from many of the challenges of definition that AI presents while providing us with a solid conceptual framework to guide us throughout.

A simple way to describe agent-based software engineering is that it is a combination of software engineering and AI. It studies how AI practitioners have been building software and attempts to identify common traits and patterns to inform the practice of building AI applications.

It concerns itself with how intelligent programs (agents) can be structured and provides ways to model and reason about the behavior of single agents and the interactions between agents. Although we are not dealing specifically with software engineering in this book, the concepts and abstractions help us understand any AI technology and, crucially, the impact it can have on our processes.

A key reason for taking an agent-based view is that we can consider *what* the agent is doing without having to consider *how* it achieves it. In other words, we don't need to wonder about what specific AI technique the agent is using to achieve its task. Too often, discussions get lost in the details of what neural network or what statistical technique or what symbolic logics or, worse still, what programming language is being used and whether that counts as AI or not. This allows camps to be formed about what is "true" AI and what is not. Very often in these cases, "true" AI is whatever technique the person claiming truth is most fond of or familiar with, and everything else is somehow lesser.

An agent-based view of AI applications allows us to consider ***what*** the application is doing, without having to concern ourselves with ***how*** it is achieving it.

While we will look at some of the basics of these approaches, in general we shouldn't care *how* the problem is solved. Technologies evolve and ways to solve problems change. If you follow AI developments, even tangentially, you soon realize that every day, week, and month brings forth new announcements and new amazing architectures. Trying to keep track of it all, unless you are a practitioner in that specific subfield, is a losing game.

We should always distinguish the *what* from the *how* and focus on the important aspects for us. If you are researching neural net architectures, solving a problem with neural nets is the important aspect. If you simply want to know if there is a cat in a picture, the most important aspect is that you get reliable answers. How you get there is secondary.

The other reason an agent-based view helps is that we don't need to actually define intelligence. As we've mentioned a few times, that is not an easy task anyway. If every time we talk about the behavior of a system, we get distracted by discussions about whether it is really intelligent or not, we will not get anywhere anytime soon. We will, instead, focus on what agents are trying to achieve and their interaction with the rest of their environment (including other agents).

Agents

One of the classic textbooks on AI[3] describes agents as "anything that can be viewed as perceiving its environment through sensors and acting on the environment through effectors." It helps if you think of it as a physical robot (Figure 2-1). A robot will use sensors (cameras, GPS, etc.) to figure out where it is and what is around it, and based on that it will cause its motors (its effectors) to act in a way that will take it closer to where it needs to be. Crucially, *how* the robot decided to go left or right is not important at this point. The internal process that led from sensing to acting does not need to be known to describe what the robot is attempting to achieve.

[3] Stuart Russel and Peter Norvig, *Artificial Intelligence: A Modern Approach (Pearson,* 2010).

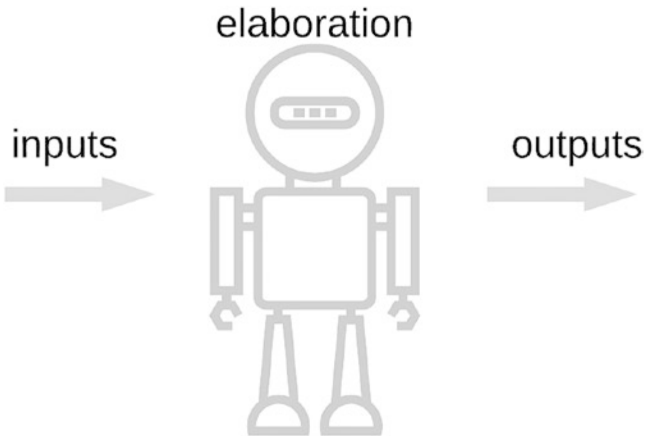

Figure 2-1. Agent as a robot receiving inputs and effecting change

Of course, the robot is moving in a certain way because it is trying to achieve something. There is some *desirable state of affairs* that it is trying to bring about, such as exit a room or transport an object. We call this desirable state of affairs its *goal*.

A **goal** is a desirable state of affairs in the environment—a state that we can describe using attributes of that environment.

Goals are crucial in defining agency. It is the *why* that drives *what* we are doing. Why did the robot turn left? Well, it is trying to get to an object that is to its left, and so on. As such, the definition of an agent for our purposes is that an agent is something that is attempting to achieve a goal through its capabilities to effect change in the environment it is in.

An **agent** is something that is attempting to achieve a **goal** through its **capabilities** to effect change in the environment it is in.

The originators of this agent-based perspective, Professors Michael Luck and Mark d'Inverno, give a conceptual and fun example to illustrate this viewpoint. In a paper titled "A Formal Framework for Agency and Autonomy,"[4] they use

[4] Michael Luck and Mark d'Inverno, "A Formal Framework for Agency and Autonomy" in *Proceedings of the First International Conference on Multi-Agent Systems*, eds. Victor Lesser and Les Gasser (Cambridge, MA: MIT Press, 1995) pp. 254–260.

the example of a simple coffee cup as something that can be an agent. A cup, they propose, can be considered an agent in certain circumstances where we can ascribe it a specific purpose (i.e., a goal). For example, when the cup is being used to hold liquid, it is achieving our goal of having somewhere to keep our coffee. Once it stops serving that purpose, it stops being an agent for us.

I was very attracted by this viewpoint of agency as I was starting out my own PhD research, precisely because it provided a solid foothold and avoided vague definitions. As a result, I further delved into their original framework in an attempt to construct a methodology that would help us not only describe agent systems but also build them.

In my work, I extended the overall framework to provide a few more categories that would allow us to more easily describe software as agents. The coffee cup is an example of what I called a *passive* agent. It is passive because it has no internal representation of the goal it is attempting to achieve. In fact, it is the *owner* of the coffee cup who recognizes that the coffee cup can play a useful role because of its capability to hold liquid in one place, and it is the owner of the coffee cup who knows how to manipulate the coffee cup (pour liquid in, hold it upright, and raise it to take a sip).

A passive agent has no internal representation of a goal. It is left entirely to the user to understand how to manipulate the passive agent's capabilities in order to achieve a goal.

Passive agents are our baseline. It is the software that doesn't take any initiative, and doesn't do anything unless we explicitly start manipulating it. While this may sound like a purely theoretical concept with little practical application, it actually describes the majority of software we use right now. Most applications are not actively aware of what we are trying to achieve. They provide us their myriad menus and submenus, buttons, and forms to fill out and it is left to us to understand how we should manipulate them to achieve our goals. Setting this baseline and having it as a starting point that describes most software today gives us something to build on to describe software that is powered by AI techniques.

Active Agents

Now, let us take it a step further. If we have passive agents, it means that we must also have active agents. Active agents are those that do have an internal representation of a desirable state of affairs. They are *actively* trying to achieve something. The simplest such agent could be a thermostat. A thermostat can sense the temperature of a room (perceives the environment) and switches off the heating when the desired temperature is reached (causes a change to the environment).

■ An active agent has an internal representation of a goal and uses its capabilities to achieve that goal.

Active agents put us on the path of the type of AI we are interested in, where we are delegating decision-making to machines. In this case, we are letting the thermostat decide when to switch the heating on or off in order to get to a desired temperature.

"Hold on there," I hear you say. "A thermostat is AI? I thought AI is about hard problems!?" That is a very sensible statement. Remember, however, that we are trying to build a framework that will help us deal with all sorts of situations without any vague assumptions of "complexity." Do not get hung up on "hard problems" or "complicated situations." Everything lies on a continuum, and we will get there in time. The important thing about the thermostat is that it perceives its environment and reacts to changes in the environment by switching the heating on and off. What helps it decide exactly when to react may be a dumb switch ("`if room temperature greater than 25 Celsius switch the heating off`") or it could be the most finely tuned neural network model that has learned the percentage by which it should increase or decrease heat output so as to maximize energy usage while optimizing comfort. Remember, we don't care about the how, just the what.

From a business process perspective, the most valuable thing is that the task of managing the temperature has been delegated to an automated process. It is no longer required that a human being keeps checking on the temperature and, based on that, decide whether the heating should be switched on or off or the heat output increased or decreased.

■ From a business perspective, active agents or active software indicates tools that we can delegate decision-making tasks to.

This is perhaps the single most important lesson of this chapter. Deciding to use AI in a work environment is deciding to delegate a decision-making task to a software program. At its core it is no different than the decision most companies make without blinking about delegating key aspects of financial forecasting to spreadsheets or key aspects of resource planning to ERP (enterprise resource planning) software. In plain words, AI is a way to achieve automation. Where AI differs from a lot of the existing software is that it broadens the scope of the type of tasks we can automate. It does that because AI technologies allow us to build software programs that are increasingly better at identifying what the correct decisions are in scenarios where it was not previously possible.

The introduction of AI in a work environment is the process of identifying what decisions can be delegated to a software agent.

When we introduce AI in an environment, we introduce software agents that make decisions in an automated fashion. In this section we talked about the thermostat acting as an active agent and how it could either be a very simple device or a more complex system. The agent is moving toward the same goal (regulate temperature) but with differing capabilities. We call the ability of an agent to vary how to employ its capabilities to achieve a specific goal *self-direction*.

Self-Directed Agents

To better understand self-direction, let us consider the notification system on our phones. You can conceptualize it as a software agent whose task is to receive a notification (input from the environment) and pass that message on to you (effecting change in the environment that achieves its goal).

A simple version of this notifier agent is one where every time a message reaches your phone (for any application on your phone) it simply gets displayed on the screen. That is it. I am sure most would agree that as far as AI goes, this is definitely on the lower end of the spectrum.

Now consider an agent that, when the message arrives to your device, refers to your notification preferences, considers your current context (e.g., is it past a certain time, are you at a certain location, what type of message is it, who is sending the message?). Based on all of that, the agent decides the most appropriate course of action (e.g., show on screen, play a sound, show message on a different device like a watch). Hopefully, you agree that there are plenty of opportunities for "intelligent" action in this scenario. There is an active consideration of the situation, which can lead to a number of different outcomes.

Both agents serve the same purpose. Namely, notify the user of messages. Some agents, however, may perform the same action irrespective of current or past behavior, user preferences, or context. Other agents make a number of decisions and have a number of choices as to how to achieve their goal. The software program stops being a dumb pipe and turns into an active participant in the process. That is why we call this ability to vary action and outputs based on internal decision-making *self-direction*.

■ Self-direction is the ability of an agent to vary the ways in which it achieves its goals.

The more self-direction we require of an agent, the more AI techniques we will need to employ to achieve it. We need the ability to reason about the world, review potentially large amounts of data, and decide which action to perform based on all of that.

Even then, it is not a simple on/off switch. Now you have AI; now you don't. Everything lies on a continuum, as Figure 2-2 illustrates. The more complexity we introduce into this decision-making process, the more contextual and historical information we need to take into account, and the more AI techniques we will have to use.

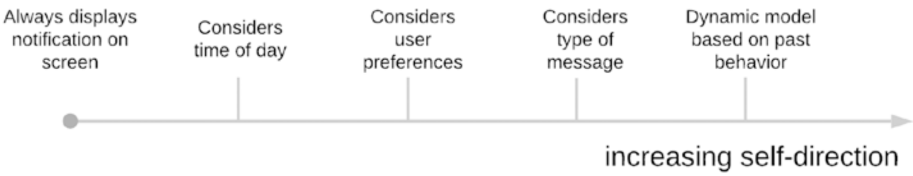

Figure 2-2. The self-direction continuum

What is essential at this point is to appreciate that from a certain high-level process perspective, it is not a question of differing levels of complexity. From a process perspective, the task has simply been delegated to a software agent. There is a piece of code that is responsible for notifying us when a message arrives. We have delegated that decision-making to software, either through explicit rules and our preferences or through more probabilistic reasoning based on context and past behavior. The question then becomes what level of self-direction is necessary from this software in order to perform its task efficiently and usefully for us.

■ AI is the process of identifying and coding decision-making skills into software programs so that they can effectively carry out the tasks that have been delegated to them.

Autonomous Agents

The notifier agent we have been describing so far, at any level of self-direction, can only operate within the bounds of a very specific goal: to decide when and how to display a message to the user.

There is another level of agency that is useful to consider: one where the agents do not just operate to achieve well-defined goals but can actually generate their own goals.

Imagine your workplace has just been outfitted with the latest "intelligent" office energy management system. This system has a target of ensuring you don't spend more than 100 "units" of energy per week and that the occupants of the workplace get the maximum amount of comfort out of those energy units. To achieve this target, it references all the available data, preferences, and rules around what constitutes efficient energy use and comfort and begins taking action. It begins formulating specific goals (desirable environmental states) that it wishes to achieve.

For example, it may decide that it should switch off certain devices because they seem to have been forgotten—switched on but not actively being used. It may also decide to just ever so slightly drop the office temperature so as to conserve energy. These are different goals that stem from its attempt to meet its higher level target. This is software with its "own" agenda and goals that is using whatever capabilities it has in order to fulfil that agenda.

Let us look at another example. Suppose you have a "wellness" agent whose target is to ensure that all members of a team get a chance to participate in company activities. This wellness agent is given certain capabilities such as access to and the ability to reason about people's diaries, or the ability to map out relationships based on interactions through e-mails or in chat software. Using that information, it can then decide to act based on its findings. It will have to make decision such as:

"Do I move that project review meeting and affect the schedules of five people so that Alexis can join a yoga session, or do I have Alexis stay past a certain time in the office to do yoga?"

These are different goals driven by a higher level target. We call these higher level targets *motivations*, and the ability of agents to pick a goal in order to satisfy their motivations *autonomy*.

Autonomous agents generate or choose between different goals, using higher order motivations.

Autonomy describes an agent's ability to vary its decisions about what goal to achieve. As with the other concepts discussed so far, autonomy lies on a continuum. Take, for example, the wellness agent from before. We said that it can monitor diaries to ensure that everyone is participating adequately in social activities. What should it do, however, if someone is not participating in social activities? This will depend on how autonomous the agent is. It could, for example, simply notify the line manager of the person in question to highlight the issue and leave it at that. Alternatively, it could decide to change an employee's work schedule so that the employee can then take the opportunity to book some time for a social activity. It could change the work schedule *and* book the social activity without asking anyone's permission.

As we introduce AI in our work processes, we need to carefully consider exactly how much autonomy we are providing. More autonomy means we are delegating more decision-making power to computer software, and we will potentially reap more efficiency out of it. It also means that we may suffer and have to deal with unintended consequences.

An informative example of this is the now infamous Microsoft Tay chatbot released on Twitter. Tay was given considerable autonomy in terms of what messages it could produce and that led to an embarrassment for Microsoft. As trolls "taught" Tay racist and extremist phrases, the chatbot used those phrases in interactions with other people. Microsoft had to recall the bot, blaming a "vulnerability"— the vulnerability was that Tay had no constraints on what it could say and no guidance as to the quality of what it learned.

Before moving on, I want to reiterate the different levels. We will use this terminology throughout the book, so it's useful to make sure we have it all well laid out.

- *Agents* are software programs that have some *capabilities* and can effect change in their environment through those capabilities. The desired change is called a *goal*.
- *Passive agents* are ascribed goals by their user. It is the user that manipulates a passive agent's capabilities. Most software we use behaves like passive agents.
- *Active agents* have an explicit representation of a goal to achieve and can manipulate their own capabilities to achieve a goal.
- *Self-direction* refers to the agent's ability to vary the ways in which it achieves a goal.
- *Autonomy* refers to the agent's ability to choose between different goals, in service of a higher order target or *motivation*.

Agents That Learn

So far, we have talked about agents that are passive, active, and even autonomous. Another key dimension is the agent's ability to learn from past experiences.

Returning to the wellness agent, we can envision how, as it tries different strategies with users to get them more actively involved, it "learns" which strategies work best with which users. This adaption of its behavior to different contexts based on previous action and historical data is what we will consider as learning.

There can be any number of layers of complexity hidden behind this learning activity. The ways the agent assigns scores to different reactions and outcomes from users can become increasingly more sophisticated. It may be able to draw not just from the reaction of a single person but that of thousands of users over a long period of time. If the data grows and the variables to consider are multiple, it will need special tools and increasingly more sophisticated AI techniques to make sense of them.

Agent Communities

For the sake of completeness, let us also briefly consider multiple interacting agents. This is an aspect of automation that is not often discussed but is crucial. No problem can be solved by a standalone piece of software. We always need to interact and integrate with other components in order to achieve our goals. This trend will only continue to accelerate.

In the near term, it is more likely that an autonomous piece of software, for example, your Siri-like phone assistant, will interact with passive services (e.g., using a timetable API to understand when the next train is coming). However, it is reasonable to assume that this will change. We will get to the point where multiple autonomous software programs will regularly interact in our daily lives and make decisions for us. At that point, we not only have to deal with how single agents arrive at a decision but also what are the *emergent* behaviors of multiple agents.

It might be a group of autonomous vehicles distributing themselves in a road network or two meeting booking agents negotiating based on their owners' agendas. Returning to our wellness example, assume that there is an agent community with the common goal of "get employees to interact more within the company." Each individual agent, however, has differing capabilities and motivations. One agent is the Social agent with a particular focus on social activities, while a Health agent is more concerned with helping users maintain a healthy lifestyle in and out of work. The Social agent may suggest that a user should take one less trip to the gym and spend that time instead doing a more social activity. The Health agent would then have to weigh the pros and cons of this. Perhaps they even enter a negotiation to decide how to settle the issue. They might settle on setting up a game of tennis to satisfy both social and health needs!

A lot of research in agent-based computing focuses on how to get agents to coordinate to collectively solve problems, and how we can reason about the behavior of a group of agents. The more heterogeneous the types of agents interacting, the more complex the problems can become.

Moving Beyond Intelligence

In this chapter we explored the very concept of AI, provided some examples of the challenges around trying to pin it to any single definition, and also distinguished between general AI and domain-specific AI.

We then referred to agent-based computing as a source of a conceptual grounding for thinking about AI, with a focus on the observable behavior and not the specific techniques that will allow us to create those behaviors. This grounding allows us to consider the task at hand, the delegation of decision-making to machines, without having to make explicit reference to notions of intelligence. Moving past vague notions of intelligence clarifies thinking.

At the same time, the agent-based perspective we introduced here needs some time to embed. Take an application you consider to be an example of AI and try to classify it from the viewpoint of agents. What goals is it trying to achieve? What information is it using? What decisions can it make? To what extent can it autonomously effect change? Does it interact with any other applications? In what ways does it interact?

I find these sorts of exercises extremely useful to start reasoning about problems and the extent to which we are really delegating decision-making power and automating. The more confident you become in analyzing problems through this lens, the more adept you will be in talking with AI practitioners about what you need to see happen.

Your business goal is unlikely to ever be to have an application built that uses the latest neural network architecture.[5] It will hopefully be defined according to a specific problem you are facing in the workplace and associated to clear objectives about how to improve the way things get done. That is the *what* we most care about. Concepts such as active and passive agency, self-direction, and autonomy help us capture the what in clear terms across a range of different domains. The *how* comes next and will be the subject of the next couple of chapters.

[5] Unless you are looking to raise VC money, that is!

CHAPTER

3

Building AI-Powered Applications

What does it mean to build an AI-powered application? In the previous chapter we started shaping our thinking around the types of behavior that AI software may exhibit, such as the proactive accomplishment of goals and autonomous goal setting. We did not, however, discuss how such behavior is achieved. We purposely did not refer to any specific technology. Technologies, of course, do matter. Technologies, though, are also constantly changing. That is why it was crucial to be able to think about AI applications without reference to specific technologies. At the same time, we need to be able to consider what technologies may be required in order to make informed choices. This chapter begins to lay the foundations in that direction. It digs deeper into the question of how AI-powered applications are constructed, and it attempts to do this in a way that hopefully anyone should be able to follow.

AI is an incredibly fast-moving space; the buzzwords and trends of today will not necessarily be the same ones of tomorrow. We could do a deep dive into the most fashionable machine learning techniques, talk about the finer details

R. Ashri, *The AI-Powered Workplace*,
https://doi.org/10.1007/978-1-4842-5476-9_3

of the latest natural language processing developments, or spend countless hours over the hottest vision developments. It would be a losing battle. By the time this book is in your hands, the content of such a discussion would already be obsolete. As such, we will focus on some of the underlying core concepts that are more likely to outlast any specific technique, architecture, or tool.

We will take a broad view, starting from the fundamental question of what any AI technology would need, and present a framework that divides things into techniques, capabilities, and applications and allows you to navigate between those three different areas and draw connections, to help you make better informed choices.

What AI Needs

In the previous chapter we referred to AI as the process of delegating decision-making to machines. Let's work through an example in this section to uncover what it is we might need in order to achieve such delegation.

You have been tasked with introducing automation to your company. A review has taken place, and it has been decided that it would be worthwhile to automate the process of evaluating expense claims submitted by employees. Software is now to decide whether the claims should be accepted or contested. It is a task that the finance team hates, but it is necessary to ensure that expenses are kept in check and only the right things go through. You don't know how to get started, so you decide to go talk to the two people in the finance team who deal with this activity.

The first person you talk to, let's call her Mary, says that they have a very precise process they follow. Mary looks at each incoming receipt and categorizes it as travel, entertainment, education, etc. Mary then considers the totals spent as well as the individual line items and cross-references those to what the person submitting the receipt was supposed to be doing at the time, as well as any rules surrounding their and their team's maximum spending levels across categories. If all rules pass, Mary will approve the expense.

The second person, let's call him Donovan, stands back in amazement upon hearing that description and says: "Wow! That is incredibly thorough, but I find that the rules change so often or are so unclear that I can't keep up. What I do is size up the person submitting their expenses and make a quick judgement call about their overall reliability and the overall amounts that come in. I also keep a record of who tends to have complaints at year-end audits about their expenses or not. If they generally don't cause trouble and I think they are reliable, I just assume that they are doing the right thing. If they tend to get into trouble often, I will push back and have them double-check themselves, just in case."

"Hold on!" says Mary in shock. "But that way you can end up doing duplicate work or wrongly categorizing things." "Yeah, I won't lie," answers Donovan. "It can happen and some people will complain, but I save so much time not worrying about every little detail that it all ends up being efficient in the end."

You go away and think about what you've heard. Mary has a very precise set of rules that she applies and can exactly explain how every decision is reached. Donovan seems to depend on past data and past behavior to make a quick determination about whether they should investigate further or not. While Donovan might miss a few things or end up with some duplicate work, the overall result is very efficient. Mary relies on having a clear as possible understanding of the governing rules at any given moment, while Donovan feels the rules are never clear enough, so he instead relies on the data going in (expense claims) and out (audit results) to make judgement calls.

Where does this leave things in terms of how you could build your automated system? It needs to make decisions. That much you are sure of. It seems that you can either explicitly list all the rules, like Mary does, or you can somehow teach it to look at past data and use that to base its decisions on, like Donovan does. Your AI system needs a way to make a determination, based on some inputs, about what the appropriate outputs are. It needs a way to reason about the world, but it turns out there are different ways of doing this.

Well, you are in good company. This is exactly the conundrum that AI researchers have been grappling with for decades. The first approach is what we would describe as a model-driven approach. There is a clear model of how the world behaves, with explicit rules and regulations that govern it. We collect all the data input points, pass them through our set of rules, and make a determination. The second approach is a more data-driven approach. It recognizes that often we simply can't explicitly list all the rules. All we know is what data went in and what data came out of the system, and we attempt to build programs that replicate that input/output relationship. We will not be able to explicitly articulate exactly why an expense was accepted or contested, but we trust that data will guide the way.

There is, perhaps, a third way to solve the expense claim problem. We could use a combination of the two approaches that tries to benefit from the efficiency of the data-driven approach while retaining the clarity and reliability of the model-driven one. Before we look at how we might do that, though, let us explore model-driven and data-driven AI a bit further.

Model-Driven and Data-Driven AI

There is any number of borders one can draw around AI work. From formal logics to statistics, evolutionary or emergent behavior approaches, and the wild and crazy world of deep neural nets, the ways you can describe and combine AI technologies are endless. Perhaps one of the most important and long-lasting distinctions, however, is that between data-driven and model-driven AI.

Model-Driven AI

Model-driven AI describes techniques that depend on explicit descriptions of a domain, the relationships between entities in that domain, and the rules that govern the overall behavior.

This approach is particularly relevant when a domain requires deep expertise that can be expressed in definitive rules. One example of a model-based system is the use of Newtonian classical mechanics to model movement in the real world. We already have a clear set of equations that can help us predict in which way the application of force on a physical object will cause that object to move given the surrounding conditions. There is no need to seek some alternative way of learning what will happen. Since we have a good enough model of what happens in the world, we can use that.

SO NEWTON'S RULES ARE AI!?

I am well aware that I keep challenging what you may have typically considered as AI. That is precisely the point.

AI is no single thing. Just like math, it is a collection of techniques. The objective is to enable us to build machines that can help us make useful deductions about the behavior of real-world objects and automate decision-making. It's best not to limit the range of possible solutions to any single technique or group of techniques.

Imagine a world where Newton's laws where not understood but, amazingly, data-driven machine learning was deeply understood. Equipped only with the tools of neural nets and data analysis, you can picture scientists recording hundreds of thousands of runs of apples falling from trees to collect data in order to correctly train a predictive neural net model of how the apples would bounce off the ground. Newtonian physics manages to do it with a small set of easy to understand equations. This is precisely the difference between model-driven decision making and data-driven decision- making.[1]

[1] As a further, small side note, in case you are curious to see if neural nets could make such predictions, there is actual work in that direction. In a paper called "Discovering Physical Concepts with Neural Networks," researchers explored this idea exactly. Raban Iten, Tony Metger, Henrik Wilming, Lidia del Rio, and Renato Renne, "Discovering Physical Concepts with Neural Networks," https://arxiv.org/abs/1807.10300, 2018.

There are many domains in which we need experts to describe the reasoning behind decision-making processes. Consider the task of making a diagnosis for many diseases. Although data plays a key role, it needs to start from a knowledge base of medical understanding that the system can use. Companies such as Babylon Health and Ada, which are changing the healthcare landscape by offering large-scale automated primary healthcare, employ specialists to build up their models of the medical domain.

Model-based AI is also essential when you need a clear path from inputs to the decision the AI system took. As we will discuss further on, one of the biggest challenges of heavily data-driven systems is the lack of transparency of how the system went from inputs to outputs. With model-driven AI, that process is often much more explicit.

Where model-driven AI fails us, however, is when we have to deal with situations wherein we cannot explicitly state what the decision-making process was that we ourselves used to get to a result. Try this experiment: what set of rules would you use to build a system that can recognize a cat whenever it sees one?

Using model-driven techniques, we might start by saying that a cat is a four-legged animal (what are the rules for describing an animal?), with two eyes (what are eyes?), a nose and a mouth. A cat is furry (except when not) and kind of medium-sized (except when not and, by the way, compared to what?). A model-based system would look at an image, deconstruct it into lines and shapes and colors, and then compare against the set of rules we've supplied about how lines and shapes and colors combine in the world to give us different animals.

As you can probably already guess, this approach falls apart quickly. There are so many rules to describe and so many edge cases, that a model-driven approach is not sufficient. This is where data-driven AI steps in.

Data-Driven AI

With data-driven AI, instead of describing rules or providing straightforward mathematical formulas, we build systems that can derive the appropriate rules themselves. We essentially take a step back from trying to model a solution and build a system that can discover a solution on its own. It does this by analyzing large amounts of data that (most frequently) has already been annotated with the "correct" and "wrong" examples of the thing we are trying to learn. The principle is incredibly simple, but the resulting systems are some of the most complex we have created.

In order to train a system to correctly identify cats, we "show" it a large number of pictures (with and without cats) and let it know when it guessed correctly. At every turn, the system uses a number of techniques to realign itself so that it is more likely to guess correctly again. This process is repeated until it is reliably guessing things correctly.

This is what neural networks do, and after decades of research we have reached a level of sophistication and complexity in the architecture that means that, for certain domains, we can have impressively reliable predictions.

There were two turning points along this path that exemplify the achievements of what are called deep neural networks and what they need in order to work.

Standing on the Shoulders of Data

The first turning point was about data.

In 2009, Professor Fei-Fei Li and her group released a dataset called ImageNet.[2] It is an annotated database of over 14 million objects. The dataset distinguishes specific objects and places them in about 1,000 categories such as animal (~2.8 million examples), bird (~800,000 examples), food (~1 million examples), and people (~ 1 million examples).

The ImageNet data sparked an annual competition for who would be able to build an algorithm that performed best in predicting what objects were in an image.

Up until 2011, the best performing algorithm got to about a 25% error rate. One out of four times it got it wrong. Then a new algorithm was introduced that changed things definitively.

Give Me Enough Data and the Right Algorithm and I Will Move the World

Geoffrey Hinton and his group published a seminal paper in 2012,[3] wherein they introduced a series of improvements to deep convolutional neural networks. Through these changes they achieved an error rate of 17%, a significant improvement. The neural network had 60 million parameters and 650,000 neurons. It used the work of Andrew Ng et al.[4] (which we mentioned in Chapter 1), taking advantage of GPUs' to efficiently handle all this complexity.

---

[2] ImageNet, `www.image-net.org/`

[3] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks" in Neural Information Processing Systems 25 (NIPS, 2012).

[4] Rajat Raina, Anand Madhavan, and Andrew Y. Ng, "Large-Scale Deep Unsupervised Learning using Graphics Processors" in Proceedings of the 26th International Conference on Machine Learning (ICML, 2009).

This new approach marked a qualitative step change. Over the next few years the error rate went down to just a few percentage points. Some hailed that as neural networks getting better than humans (since we get it wrong about 5% of the time). Cooler minds reminded people that the competition only needs to be accurate across 1,000 categories, whereas humans can recognize a much bigger range of categories as well as complex context.

The point, however, is that deep neural networks finally got us to a point where we could do interesting things reliably enough, in ways that model-based AI cannot support.

The deal does not come without costs. We sacrifice a lot of clarity and explainability along the way. We don't actually know why those 650,000 neurons light up in a particular way (and neural nets nowadays can be orders of magnitude bigger). There is no chain of decisions to point to. We don't know what features from the input data are the ones that are determining specific choices. This can present important ethical and legal challenges, as we will explore in subsequent chapters.

Techniques, Capabilities, and Applications

As I am sure you can begin to see, there is a dizzying array of techniques and approaches to solving the problem of delegating decision-making capabilities to software.

In order to try and bring some order to thinking and rein in all the disparate approaches under a simple model, I break down the task of building AI-powered applications into three parts.

It starts with the **techniques** that enable us to reason about the world either by allowing us to explicitly **model** an aspect of intelligence or by helping us **discover** it**.** These techniques are divided into model-driven or data-driven.

Techniques combine to provide **capabilities.** Capabilities are specific systems that empower us to understand an aspect of the world or to effect change. Capabilities are things such as machine vision, speech processing, and natural language processing. Whether model-based, data-based, or a hybrid combination, a capability is a "super-power" that a machine has that enables it to do something.

Finally, capabilities combine to give us complete **applications**. An application is the finished tool to which we ultimately delegate decision-making powers. It is our expense claims adjudicator, our chatbot virtual assistant, our financial markets predictor, etc.

The relationships between techniques, capabilities, and applications are illustrated in Figure 3-1. To me, this provides a clear way to differentiate between different levels of work within AI. Obviously, within a business setting we ultimately care about the applications that are enabled. For those applications, we need specific capabilities—specific ways of understanding and changing the world. These capabilities are made possible because, at the level of AI, researchers are working on a whole host of different techniques from logic-based reasoning to data-driven deep learning architectures. Identifying where to focus while not losing sight of the bigger picture is crucial. Too often, people are stuck on a single technique (typically neural networks because that is what gets talked about most) and forget that what really counts is a solid capability or a finished application. Vice-versa, a business may be calling for an application to be built without realizing that the necessary capabilities are simply not there, so they would need to be investing in primary research to develop the techniques that could lead to the necessary capabilities.

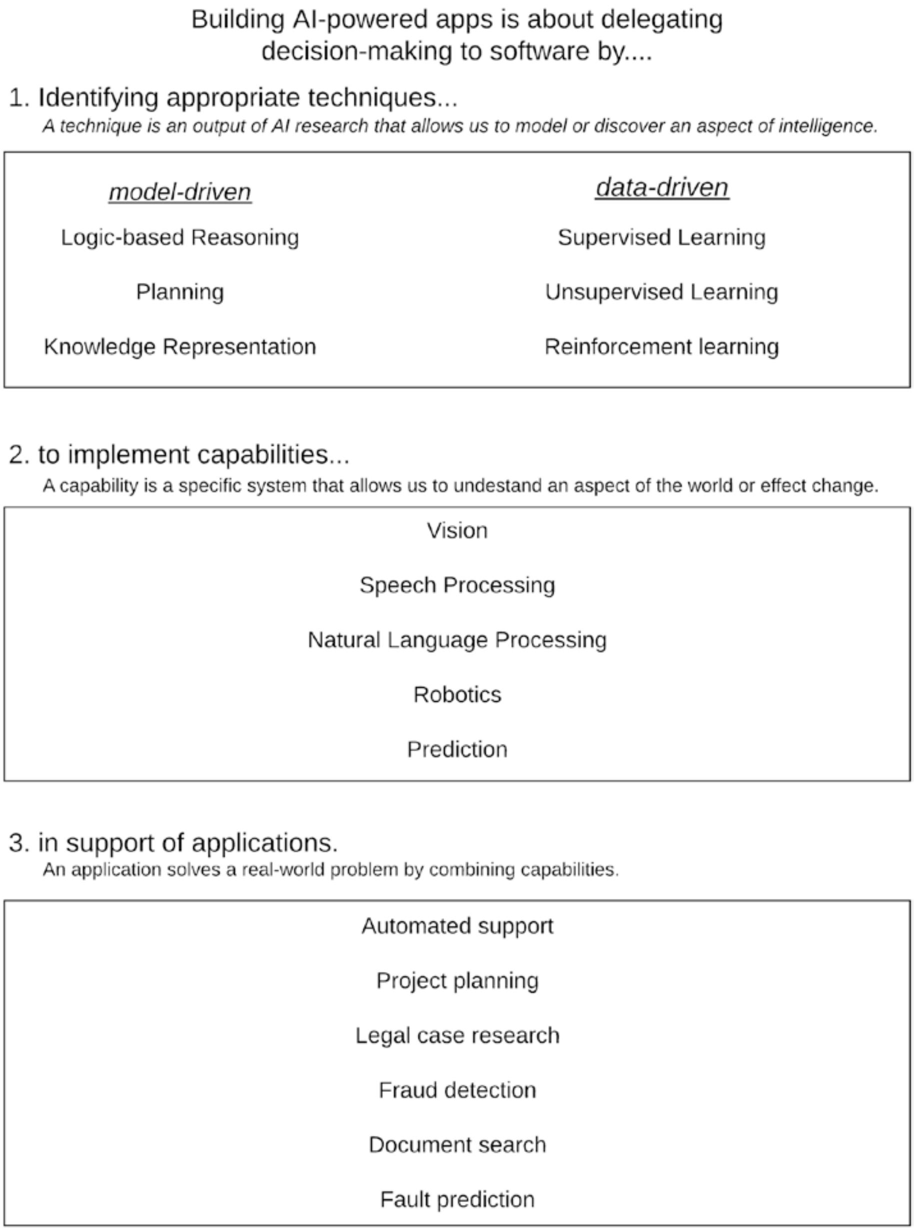

Figure 3-1. From techniques to capabilities and applications

Experts and Machines Working Together

In the next couple of chapters we will look more closely at some of these capabilities and techniques, but before we dive in, let us return to the application we have to build for this chapter: our expense claims adjudicator.

Ultimately, what we need is a combination of approaches. We need model-driven knowledge representation that encapsulates the core rules and regulations. We can then back that up with data-driven decision-making that references past data to uncover patterns that our explicit model does not capture faithfully enough.

Real-world AI applications are almost always a combination of a model-driven approach together with a data-driven approach. Mapping technology is one of my favorite examples. Consider the Google Maps application on your phone. On one hand, you have a very explicit representation of roads, buildings, and the rules that govern how you can get from A to B based on road signals, speed limits, etc. On the other hand, Google Maps relies heavily on data-based predictions to reason about what is the best course to take given the time of day, inputs from other users, etc.

Every project needs to ensure that it understands the problem and objectives, takes into account the knowledge of experts who deal with the subject every day, and formulates a plan about how to make best use of all the tools available. If an explicit model can be derived, it can provide the most efficient path from question to answer. Sometimes, however, the features are so many and the process so unclear, that only a data-centric approach can work.

At each step, we must be realistic about the feasibility of either approach. Is it a fool's errand to try and build a model of a situation that is simply too complex to describe with rules? How much data will we need, what is the state of the data we currently have, and how reliable will the end result be? How can we prove to ourselves that we are heading in the right direction? How can we build fail-safes in the solution by combining techniques?

Over the next few chapters we are going to delve into these questions as we explore more specific technologies and discuss how you can formulate a strategy that exploits AI for your workplace.

CHAPTER

4

Core AI Techniques

How does one even start to capture the processes that lead to what we recognize as "intelligent" behavior? More appropriately for our definition of AI, how does one begin to program machines so that *they* can make decisions instead of us?

As we already discussed in Chapter 3, there are two broad approaches. We can either develop explicit models that govern the behavior of our system or we can attempt to discover those models by analyzing data and looking for patterns. In this chapter we provide a very high-level overview of what the main techniques are. Understanding the thinking behind these core techniques enables you to reason about the pros and cons of different approaches and the needs they might place on your products and teams.

Artificial Intelligence is a fast-paced field and it feels like there are "new" things every week. This can make someone think that it is pointless trying to "catch up"; that it is best to leave everything to the experts. While it may be true that there are constant developments in the field, and setting yourself a goal of being always up to date is a losing battle, it is equally true that the core concepts have been around for decades. Whether it is semantic knowledge modeling or artificial neural networks, the ideas have been around since the 1960s. It is the specific algorithms and approaches that have evolved in the

R. Ashri, *The AI-Powered Workplace*,
https://doi.org/10.1007/978-1-4842-5476-9_4

meantime. Having a clear understanding of the overarching concepts is what will provide the most value in the long term and will allow you to reason about what appropriate strategic direction to take.

Model-Driven Techniques

Model-driven techniques are a celebration of human ingenuity. It is us looking at the world and inside our brains, identifying the core pieces that are important and connecting them in a way that allows us to make predictions. Because they are the result of our own mind elaborating on concepts, we can also fully understand them and explain them to others, something that is incredibly valuable. When a model-driven technique is correct, it is often the most efficient path through a problem, as it captures just what it requires and allows us to build on solid foundations. Think of the core rules of thermodynamics, chemistry's periodic table, or the three laws of biology: incredibly powerful statements that govern large swaths of how nature behaves.

The model-driven techniques we review in the following have been actively used in applications for decades now. They are the techniques that have allowed companies to optimize the coordination of their transport fleets, led to better search results on the Web, improved manufacturing processes, helped cure more people, and so much more. They are not the techniques that people typically refer to when talking about this current moment of AI renaissance, but they are nevertheless crucial in building sophisticated applications that can efficiently and robustly make decisions.

In looking at model-driven techniques, we will examine three core aspects: how to represent information, how to reason over it, and finally how to plan.

Knowledge Representation

The goal of knowledge representation is to provide us with tools and techniques to describe data in a way that allows us to more easily manipulate it with computers. That refers both to single items (e.g., how to describe a stand-alone document) as well as the relationship between items (e.g., how does a specific document relate to a project).

No matter what organizational context you operate in, you undoubtedly produce documents about meetings, proposals for clients, project reports, and so on. Typically, you will have some sort of document management system where all this data gets stored. Maybe it is as simple as a shared hard drive on a local network or a shared Dropbox or Google Drive environment. Now, imagine that all the documents were dropped in the same folder, called "OUR STUFF" and were all named in inconsistent ways:

- OUR STUFF
  - sales-rep.doc
  - client-abc-preso.pdf
  - my-cool-thoughts-on-stuff.pdf
  - projections.xls
  - …

As the documents grow to hundreds and then thousands, a significant amount of organizational effort would be going into simply searching through this unmanageable folder to find something.

Hopefully, in your organization things look a bit more like this:

- Documents
  - Sales
    - Client ABC
      - Presentations
      - Offers
      - Final Contract
  - Project Work
    - Client ABC
      - Project Reports
      - Deliverables

Simply by managing folder structure and imposing some rules around where documents should go, you have introduced knowledge representation to your team. This simple hierarchical structure makes it easier for people and machines to find information.

In general, knowledge representation is the effort to identify an appropriate model to capture what we know about the world, together with means to manipulate that model in order to infer new things.

Let us consider another simple example. Suppose you are the HR department of a very large organization and you receive hundreds of CVs daily from people with all sorts of skills. You would like to be able to automatically categorize those CVs based on the skills that people mention, so that you can contact the appropriate subject matter experts within the organization who would need to evaluate them. You have five high-level groups with titles such as front-end engineers (people who specialize in building the user interfaces and visual aspect of digital tools), back-end engineers (people who specialize in data management, algorithms and systems integration), project managers, quality assurance and testing, and site reliability engineers (the ones who make sure all systems run smoothly).

However, you have a challenge. The terms people use in their CVs to describe these skills keep changing and, especially, the technologies that are related to these skills keep evolving. New programming languages, frameworks, and so on are constantly being introduced. How can you automate the process of sorting through CVs in an appropriate fashion?

You get together with your team and decide that you are going to build a tool that will capture terms that are used to describe these skills and relate them to your high-level groups. The subject matter experts across the organization will be able to use the tool to enter keywords that they are interested in and then software will refer to that "knowledge" to sort through CVs. Following is an example of the type of information captured:

- Front-end engineer
    - Core Skills
        - JavaScript
        - HTML
        - CSS
    - Frameworks
        - React
        - Vue.js
        - Node.js
    - General Skills
        - Version Control
        - Testing / Debugging

Congratulations, you have just built a rudimentary knowledge graph[1] or ontology! An ontology is a more structured description of the elements that make up a certain domain, together with relationships that connect those elements together and a way to reason about the implications of certain connections.

Ontologies capture our understanding of the world and they can come in many different forms, from simple thesauri to hierarchical taxonomies like the preceding example to far more sophisticated networks of interconnected entities. Knowledge representation and knowledge reasoning techniques focus on formalizing the way we build and describe things such as ontologies, so that we can capture increasingly more sophisticated types of information in a way that remains tractable for machines to reason over. The aim is to enable us, given a set of facts, to infer additional knowledge. If I know it is furry, and it has four legs, and it makes a purring sound, can I assume it is a cat? A good ontology should be able to tell you that there is actually a range of animals (very likely all feline) that would fit that description, so you cannot simply assume it is a cat.

We now have a rich and sophisticated toolset to work with; ontologies at scale and applications can be found across a variety of fields from medicine to e-commerce. Ontologies can be created and populated "manually" by subject matter experts, but they can also be created automatically using data-driven techniques to identify and extract the relevant entities and relationships and can then be further curated by experts.

Ultimately, any sufficiently complex automated system—whether through formal means or through informal, ad hoc implementations—will end up representing knowledge in a way that machines can manipulate. As such, knowledge representation and management becomes a core technique for most applications.

Logic

Logic is at the heart of everything we do with computers. The building blocks of the processors at the core of our machines are logic gates that combine in a variety of ways to gives rise to the complex behavior we need. When we program, we typically use predicate logic to define what should happen. Consider the following one-line statement:

---

[1] In recent years, especially after Google launched what it calls its Knowledge Graph, people started referring to various forms of ontologies as knowledge graphs. The Google Knowledge Graph is what powers the answers you get to the side of Google search results in a standout box. Following an analysis of your query, if they are able to pinpoint a specific reply to your question (as opposed to a search result) they will provide that. Since then, the term knowledge graph has been creeping into literature, and you might find ontology mentioned in more formal settings and knowledge graphs elsewhere. Ultimately, it all points to the same end result: a structured representation of information that machines can reason over.

"If temperature reading is over 25 degrees Celsius, switch off heating."

This is a simple program using predicate logic to determine how to deal with heating in a certain environment. Now, imagine that in order to switch off something, say a component in a nuclear plant system, you would have to consider hundreds of statements (or propositions) that need to be satisfied and not just a single one. Further, suppose that those statements are not simple yes/no answers, that they in turn can kick off other processes, and that the order in which things take place is also important. How can you systematically step through the reasoning process and come to an outcome that allows you to take an action. This is what logic systems are all about.

Logics, in the broadest sense, concerns itself with providing appropriate formal structures to reason about different situations in the world.

There are different forms of logic that tackle different aspects of reasoning about the world. For example, epistemic logic tries to tackle the problem of what is known, and especially what is known among a group of agents that are sharing statements and beliefs. Temporal logic helps us reason about temporal events and the nature of time. It is the type of logic a machine will need to be able to reason about statements such as "My alarm will ring for 30 seconds unless I stop it earlier and will start again after 5 minutes if I press the snooze button." Deontic logic tackles the problems of what is appropriate, expected, and permitted. There are numerous other formal logic systems and combinations of them trying to capture and codify all the different aspects of life and what we as humans so effortlessly handle every day.

Logics will play a big part of providing the types of behaviors we will come to expect from more self-directed or autonomous software. Consider the following example. A user visits the web site of a car manufacturer and engages an automated conversational agent (a chatbot) to determine what sort of car they should purchase. The user, prompted by the conversational agent, might provide information along the lines of: they like to go outdoors, they have a large family, they have a pet, and so on. The conversational agent in return starts offering some options of possible cars. There's nothing particularly strange so far. The differentiation, however, comes when the prospective client rejects a proposed choice. Our logic-powered agent is able to ask why that choice was rejected. For example, the user might say: "Because, I don't think it will be able to fit all our equipment for a trip to the mountains." A simple agent would not be able to counteract that argument, and would simply move on. An agent that uses logic, though, might be able to offer a counterargument. Something like: "Well, did you consider that you can fold down the back seats or add a roof rack and carry large equipment that way?" For an automated agent to achieve this, it needs knowledge of how a car behaves, coupled with logic that will describe the effects of actions. This way it can deduce that folding seats creates more space, which is a valid counterargument to present to the user. Software that is able to offer facts and counterarguments can become a much more active assistant for us, not only helping us complete a task but also offering choices on how to complete the task.

Logics have another key role to play. As automation takes over more aspects of our lives, we will have to be able to offer more concrete assurances that they will behave in certain ways and that we can trust the decision-making done. Logic and model-based reasoning, in general, will play a large part in helping us ensure that systems are safe and that they can be trusted.

Planning

With knowledge representation we can describe our world and, using logics, we can reason about it. That is great, but how do we set in motion actions that will help us achieve a goal? This is where planning comes into play.

An easy way to conceptualize planning is to think of an actual robot trying to work out how to solve a problem. Imagine a robot that has a number of different capabilities such as moving backward and forward (or even sideways), jumping, going up stairs, picking things up and moving them around and so on. These are the *actions* it can perform to change its state in the world, using *sensors* to understand what state the world is in.

Now, imagine that the robot is told that it has to move a chair from one point in a building to another. This represents its *goal*, the desirable state of affairs. It can't just start aimlessly doing things until the chair is where it is supposed to be (which would be quite entertaining but not very useful). Instead, the robot needs to formulate a plan, potentially with intermediate goals and the actions that will achieve those intermediate goals until it completes the final goal. A plan would look something like this:

1. Locate chair
2. Move to chair
3. Pick up chair
4. Move chair to desired location
5. Place chair in desired location

Planning software would have to be able to come with this plan, monitor its progress, and replan as things change, such as someone moving the chair from its original location or something getting in the way of reaching a destination.

> Planning is the process of identifying what actions, and sequences of actions, will enable automated software to achieve a specific goal given its current context.

Anyone who has ever had the "joy" of having to schedule work, or plan activities or a course of action is painfully aware of what a daunting task this can be as the number of activities increases, interdependencies emerge, and you are constantly having to replan. Planning techniques allow teams to handle large pieces of work and ensure adherence to constraints across thousands of individual items with complex constraint reasoning and hundreds or thousands of rules. Commercial software using these techniques plays a key role in building bridges, launching spacecraft, and producing airplanes.

Data-Driven Techniques

Discussions about data and what it can enable occupy the overwhelming majority of thinking in the AI techniques space. Visionary statements abound of how every action can be measured, stored, and then used to predict our desires, needs, and intentions and influence our next action. Sometimes the use of data feels almost child-like in its naïve simplicity. You liked a picture of a friend riding a bicycle? Here are ads so that you can buy a bicycle yourself! Visited a site that sells shoes? We shall "retarget" you and inundate you with ads from that very same site for the next few weeks (quite often even if you actually already bought those shoes from that very same site!). Have you walked enough steps today? If not, we might need to give you some gentle "nudges" tomorrow so you can catch up.

It's easy to be cynical about the data age (especially if, like me, you tend to be cynical about most things!). However, it is important to not underestimate just how important data-driven techniques are for us now and in the future. While model-driven techniques can give us certainty and safety and demonstrate how human intuition can cut through the noise and focus on just what is really important with fundamental rules, data-driven techniques release us from the limitations of what our own mind can discover and give us the superpower of being able to create something without having had prior knowledge of how to create it. We build machines that explore and create for us. Perhaps within one of these machines there will eventually be large portions of the answers we so desperately need to fix our climate and heal our bodies.

In considering data-driven techniques I decided to avoid going through a long list of all the various architectures and approaches that are, anyway, in constant evolution. The Web is awash with information, and if one wants to delve more deeply, they can easily find a lot of great examples. Instead, what I want to highlight are the three core approaches and their relative differences.

We will look at how we can discover models through machine learning using supervised, unsupervised, and reinforcement learning. The one exception to this rule is a brief look at artificial neural networks and deep learning. Strictly speaking, they could be categorized under supervised or unsupervised learning but since they are so often referred to, it is worth addressing them directly.

Supervised Learning

The bulk of applied machine learning is currently focused on supervised learning techniques. Supervised learning attempts to build a model using data that is already labeled. The "supervision" consists of referring to those labels in order to indicate to an algorithm whether its prediction was correct or not. Through an iterative process during which the algorithm adjusts its decision-making process, you hope to arrive at a final model that will act as a reliable predictor.

Supervised learning refers to techniques wherein algorithms use annotated data (i.e., data with the "correct" desired answer already provided). In training, the responses are *supervised,* and the algorithm is informed on whether it got the right answer. This information is used to adjust the model.

Let's work through an example to highlight the key phases of a supervised learning process. Assume that you are tasked with the problem of renewing the company document store. After several mergers, software upgrades, and personnel changes your document store is in a mess. You know there is valuable historical data there, but you cannot sort through documents appropriately. You decide that a first useful step would be to classify those documents along broad categories that would make sense for everyone (e.g., sales documents, project reports, team evaluations).

The phases you typically would need to go through in a supervised learning process are

- Gather and prepare data.
- Choose an appropriate machine learning algorithm and fine-tune it.
- Use the resulting model from the previous phase to predict.

Let's consider each phase in turn.

Gathering and Preparing Data

You've kicked up enough dust and leaned on enough people to get all the documents in a single place. Never underestimate just how complicated it can be to simply get to the data. Departmental processes, internal politics, fear of regulation, and so many other factors can easily spell an end to your automation dreams before you even get started. It is always worth carefully planning for this phase before committing other resources. There is nothing quite as inefficient as having a highly skilled machine learning specialist or data scientist sitting around while you need to have yet another meeting to determine who you need to talk with to get access to data. You are one of the lucky ones, however. Your data is all there. All the documents are ready to be classified.

In a supervised learning scenario you need to select a part of your data and annotate it appropriately so that you can use it in training. You need to identify key features (title, summary, author, department, date, word frequency,[2] etc) that can help determine the type of document, and then you classify your data with the correct answer or target variable. Please note that each one of these decisions carries with it a complex set of considerations. Have you selected an appropriately representative subset of data? If not, then your model is not going to behave correctly with the entire dataset. You may have introduced a number of different biases, since your model is going to favor data similar to the one that was used to train it. Assuming the data is representative, have you selected an appropriate set of features to focus on? Choice of a wrong feature can once again lead to unwanted bias. The machine learning community as a whole is developing best practices in order to guard against some of these issues, but there is no fail-safe approach. It requires patience and experience, and you need to truly embrace failure as learning— you are exploring an unknown world and using software to help you craft a model that makes sense of it. Like any explorer and scientist, you need to embrace the inherent risk that comes with it. Of course, the payoff at the end of the process is huge. Having successfully automated a hard task, you give your organization a marked competitive advantage.

[2] Please note that I am simplifying here considerably. Typically, for text classification word frequencies are the key feature, and the way you represent these frequencies (as mathematical vectors rather than actual text or sums) is quite sophisticated and is a field of study within natural language processing in and of its own accord.

Choosing a Machine Learning Algorithm, Training, and Fine-Tuning

With data in hand, your next task is to determine what machine learning algorithm you should use to help you build a prediction engine. As we've already mentioned, there is a wide range of choices coming from mathematics, statistics, or computer science, and new approaches and architectures are invented at a breathtaking pace. It is the task of the machine learning expert to identify what would be the most fruitful or promising approach given your specific problem and type of data. Once more, you need to keep in mind that there is no simple answer or simple set of steps to arrive at an answer. Choosing and refining a machine learning architecture is a process of experimentation. For our text classification problem, solutions can range from something as "simple" as a Naive Bayes classifier to complex convolutional neural network architectures or to something novel that is created exclusively for your dataset. A good rule of thumb is to try the simpler approaches first and only move up in terms of complexity if you are convinced that the additional effort is justified given the potential benefit gains. The typical question is whether the cost of getting a hypothetical additional 2% boost in performance will justify the costs it will take to get there.

Training is the process of feeding the algorithm data, allowing it to adjust its various "weights" as it searches for a "combination" that will enable it to provide correct predictions. Annotated data is typically split between a training set (i.e., what will actually be used to develop a model), and a test set, which will be used to validate the model. Even here you can see how important it is to properly distribute your annotated data between the training set and the validation so that they are each a good representation of the mix of data that your model needs to be able to handle.

Fine-tuning the model (or parameter tuning as it is often called) is the process of adjusting factors that affect how the machine learning algorithm behaves, to identify what that might do to the results. You might change how many times you pass the training data through, or how significant a wrong prediction at any given point is considered and how much that should impact the change of parameters. Once more, we need to accept that this is a process of experimentation and you need to constantly be reassessing how much more effort it is worth investing in the overall process.

Predicting

Finally, with a working model now discovered through machine learning, we are ready to deploy it and do prediction on completely new data. There are two types of prediction that machine learning models tend to do. On the one hand, you have classification tasks, such as what we would need in order to classify our documents. The input document would be assigned a category based on what the model believes the document is discussing. On the other hand, you have what are termed regression tasks. A typical regression task is to predict the value of a specific item given some input characteristics, such as to predict the value of a house given its location, size, configuration, what other features it has, and so on.

Unsupervised Learning

What happens if you don't have any labels for your data? Well, to start with there are some things that you will simply not be able to teach an algorithm. You can't teach something what a cat is without actually showing it a cat. Having said that, there is a lot that algorithms can do to uncover potential correlations or groupings in our data that can teach us something.

Unsupervised learning analyzes data to uncover possible groupings or associations, without the need of any annotated data.

A typical application is to use it to segment or cluster datasets in closely related groups. Unsupervised learning can, for example, be applied to customer purchase data to identify if your customer base can be split into groups that can provide you with some insight about that group. Something along the lines of "customers who purchase product A tend to purchase product C as well" or "customers who purchase product D all tend to come from a specific geographical area."

Unsupervised learning is also, at times, used in combination with supervised learning. Under appropriate conditions, a model can be generated using training data that is a mixture of both labeled and unlabeled data. Simplistically, you can consider unsupervised learning to be doing some of the potential classification for us and that is then mixed with supervised learning. Although, in general, this should be considered a relatively risky and unreliable strategy, there is a promising growing body of research about it. In the coming years unsupervised learning will start playing an increasingly more important role, as machine learning engineers are constantly faced with the problem of having a lot of data in general but not enough labeled data.

Reinforcement Learning

Finally, we come to reinforcement learning, the fun cousin in this trio of machine learning approaches. Reinforcement learning is the closest to how one would intuitively think of training and learning in nature.

When we are training our pet to do something such as coming when called or sitting when instructed to do so, we don't present it with lots of correct and wrong examples of what coming or sitting looks like! Instead, what we do is try to coax the pet into doing what we would like it to do and once it does it, reward the pet heavily. This rewarding reinforces that this is the correct behavior. We keep repeating the process until the pet clearly associates the specific command like "Come, Max!" to the reward and ultimately the desired behavior. Similarly, and hopefully very thoughtfully, when a wrong behavior is identified we punish the pet (ideally with not much more than the use of a firm voice or a sharp look). This teaches the pet what the undesired behaviors are, because they will lead to punishments and not rewards.

These are the principles that reinforcement learning takes into the digital world. The usual setup is that some sort of environment is defined in which an agent can act, and the environment designer provides punishments or rewards when desired states are achieved. There is a wide variety of approaches researchers can take in training an agent, such as constantly providing feedback or simply providing a reward (or a punishment) at the end of a game. For example, you can have an agent playing chess, teach it nothing about how chess actually works (other than how it can move the pieces) and only provide feedback at the end of a game. The fact that the agent can run through millions of games means that eventually it might just stumble on an interesting chess strategy that leads to winning even though it started out with no knowledge of the game.

The big moment for reinforcement learning came when Google managed to build a system, AlphaGo, that defeated the world champions in Go. Go is considered a much harder problem to solve than chess, since there are many more states that the agent can find itself in, making constant calculations and searching for the next optimal move an almost intractable problem. As such, after IBM's Deep Blue conquered chess, many AI researchers turned their sights on Go. Ultimately, the team at Google DeepMind won. They used a combination of supervised learning and reinforcement learning to train deep neural networks alongside novel search strategies[3] to deliver the winning approach. Interestingly, this combination of techniques meant that AlphaGo was able to be much more "strategic" than Deep Blue, which relied on more brute force techniques of evaluating all possible outcomes of a game from a specific position. In addition, AlphaGo discovered the correct ways to play through supervised and reinforcement learning, rather than having more explicit evaluation functions provided to it.

[3] David Silver et al., "Mastering the game of Go with deep neural networks and tree search" *Nature* 2016, 529: 484-489.

Reinforcement learning is still very fertile ground for artificial intelligence research and there is much for us to discover. While winning at games such as Go is about going after the grand challenges of AI research, there are very practical applications across a number of industries from robotics to manufacturing, transport, trading, and more.

Deep Learning and Artificial Neural Networks

Deep learning (DL) and artificial neural networks (ANNs) are terms that are mentioned heavily within the context of AI, so it is worth providing some clarity here as to exactly where they fit and what they are. To start with, let us clarify that ANNs are a way to achieve mostly supervised or unsupervised learning. There are several other ways to achieve that, but ANNs are the most exciting area of development and the source of much progress in recent years.

The fundamental premise of ANNs is that the way to reach a decision is by feeding data into a network of "neurons" connected in layers.[4] Each neuron accepts an input, which it processes via an *activation* function associated to that input (an equation that, given a number of inputs, will give us an output) and will then fire off a subsequent input to the neuron or neurons it is connected with in the next layer along a weighted path. See Figure 4-1.

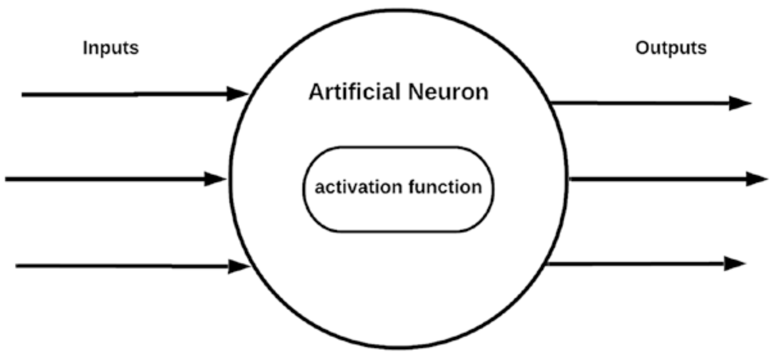

Figure 4-1. Single artificial neuron

[4] I placed the word neurons in quotes because it is important to remember that these artificial neurons have very little to do with how neurons in our brain work. While brain neurons may have been the source of inspiration for artificial neurons, we now know enough about how the brain works to at least be absolutely sure that the functioning of ANNs bears little resemblance to the functioning of the brain.

Each layer, broadly, specializes in identifying some feature of the input information and that information is fed forward to subsequent layers. There may be any number of neurons and layers internally, but it will all eventually lead to an output layer where the final set of neurons that gets activated will provide us with the answer. See Figure 4-2.

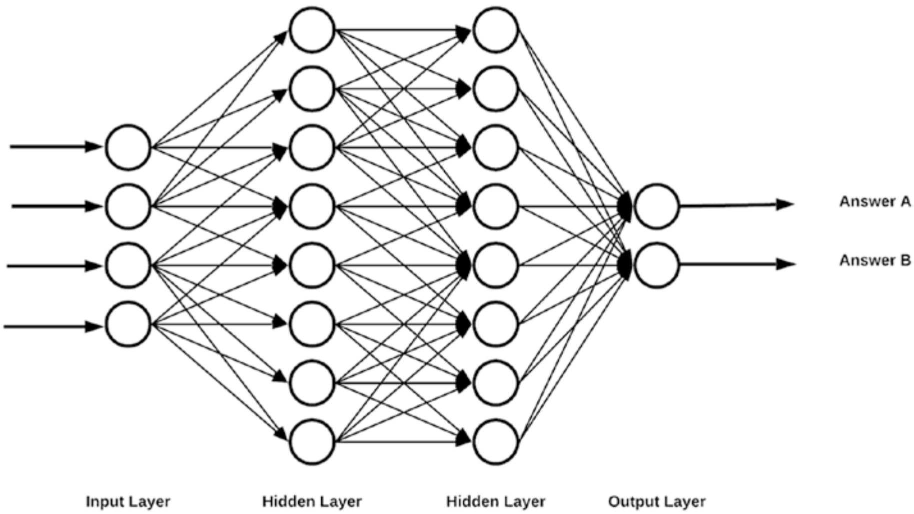

Figure 4-2. Artificial neural network with multiple hidden layers

When we train an ANN, we are using a training model to manipulate parameters (or biases) associated with individual activation functions on each neuron, as well as on the connections between neurons, until the final layer starts providing the desired results. The way these parameters change after each training cycle and how it all leads to a good result in the end is what DL experts focus on.

DL is a catch-all term for techniques that use ANNs, typically in heavily multilayered architectures where layers can be connected both forward and backward and where multiple architectures can combine to form a whole.

The main advantage of DL techniques is that they significantly reduce the need to identify what features one should input into the ANN in order to train it. For example, if you are trying to train a model to correctly recognize a face, you might start by decomposing objects in an image into basic geometric shapes and input that information into an ANN. With DL, it is the network itself that will do the work. We just input all the raw data: the value of every single pixel in the image. The ANN will extract its own features based on its architecture and the training data, with each layer "learning" a feature and the subsequent layers aggregating those features into higher level concepts.

However, and here is the catch, we need *a lot of data* in order for appropriate features to be discovered. Furthermore, the resulting network is very opaque to us. We do not know exactly what it decided to "focus" on in order to classify an image or part of an image as a cat rather than a dog. In fact, AI literature is littered with fun examples of how ANNs can get it wrong or focus on just a very small number of features that lead to very brittle solutions. This is why input data is very important. For example, suppose you want to distinguish between different objects, say cars and bicycles. If all the pictures of cars you show your ANN are of cars in a city, whereas the bicycles are in the countryside, your ANN is likely not going to work if you show it a car in the countryside. It is just as likely to use the appearance of multiple trees or lots of green as an indicator that something is a bicycle as it will use features of the object itself.

The key is to remember that although ANNs may impress us with their results, they have no semantic understanding of the data they are processing. They are simply looking for any patterns that they can use to classify input data one way or another. We quite often mistakenly attribute meaning to results and assume that our ANN has discovered something relevant to what we asked. We should never assume that. Instead, we need to thoroughly test ANNs with an appropriate variety of data and put in place governance to limit the impact of wrong automated decision-making so that there is enough confidence that the overall system will work within satisfactory parameters.

From Techniques to Capabilities

In this chapter we reviewed the core artificial intelligence techniques that allow us to develop specific capabilities. These are the building blocks, the techniques that emerge out of research labs and can then be combined and applied to give us complete systems.

The most important takeaways are:

1. The core problem is that of finding a model that allows us to describe and predict what will happen in a given scenario so we can enable decision-making.

2. We should not place limitations in terms of where that model can come from. We can explicitly design it (model-driven) but we can also use data to helps us discover it (data-driven).

3. Although this is a fast-paced field, the core concepts do not change that quickly. Even ANNs, which are viewed as cutting-edge, have been around for decades. Having a basic understanding of the key principles behind different techniques helps when picking a tool or discussing potential solutions with a team.
4. Keep an open mind about how techniques can combine to lead to a final result. When evaluating potential tools for your own problems, don't be distracted by discussions about the purity or authenticity of one approach vs. another. Focus instead on the quality of the final capability that you get.

As we will see in the next chapter and in subsequent sections, a complete application is always the combination of a number of different techniques.

CHAPTER

5

Core AI Capabilities

In the previous chapter we saw that there are lots of different techniques we can use and combine to model aspects of intelligent behavior. On their own, however, they will not get us far. These techniques only have value in as much as they allow us to do something specific and clearly identifiable: transcribing speech to text, classifying a document, or recognizing objects in an image. To achieve these tasks, we typically need to combine techniques into capabilities.

AI capabilities represent a concrete thing we can do to better understand the world and affect change within it. They are analogous to the human senses. Humans can see, hear, smell, touch, and taste. Each one of these senses involves a number of subsystems (techniques) that combine to provide the final result.

Take the ability to see, as an example. Light passes through the cornea and lens of our eyes to form an image on the photoreceptors. From there, via the optical nerve it reaches our brain to the primary visual cortex. Information there gets processed again, and is eventually mapped to specific concepts. We employ different techniques for collecting light, transforming it, and processing the results in support of a single capability: sight.

R. Ashri, *The AI-Powered Workplace*,
https://doi.org/10.1007/978-1-4842-5476-9_5

In this chapter we will focus on three broad classes of capabilities that represent the most frequent types we encounter in a work environment. They are also the most likely to provide immediate benefits in any work environment:

- The ability to understand and manipulate language (both voice and text) and generate language
- The ability to manipulate images, classify them, and identify specific objects in images
- The ability to combine organizational-specific knowledge and data to create organizational-specific capabilities—our very own *superpowers* that can be incredibly hard for others to replicate

The aim of the chapter is to give you a high-level understanding of how these capabilities work and examples of their application, so as to demystify the processes and allow you to more clearly consider how you could exploit them in your own work environment.

Language

Language is a critical capability that organizations should be looking to exploit as much as possible. As knowledge workers our currency, in many ways, is words. Whatever the end result of the activity of any office, the way to collaborate with colleagues and share ideas is through language.

Language has some fascinating idiosyncrasies and calls from the outset for a rich and interdisciplinary approach. It would be impossible to cover all the challenges here, but I think it is useful to consider a few so as to better comprehend the scale of the task and realize what an incredible amount of progress has taken place.

To start with, there are obviously multiple languages to deal with. Luckily, different languages present several similar characteristics, which means that techniques developed to handle one language can often be applied to others, with the main caveat being the availability of large enough data in the language we are looking to analyze.[1] Language, however, is not static. The English spoken in the UK today is very different from that of past centuries, and the English spoken in the United States or Australia is sufficiently different from that of the UK that different language models and datasets may be required. Language also morphs as it moves from one domain to another. If two experts in civil engineering listen in on the conversation of two experts in aerospace

[1] While languages do have some innately similar characteristics, we should be careful to not overgeneralize. A more nuanced statement would be to say that languages with similar heritage share similar characteristics.

engineering, they may understand most of the individual words but the overall meaning will be lost to them. Words take on new meanings, acronyms are introduced, and quite often, especially in spoken language, slang is used that only makes sense in very specific contexts and time periods. I am sure that if I asked my dad to "Slack me" he would have a very puzzled look, but if I said, "Skype me" he would understand and likely reply with "Why don't we just use FaceTime, shall we?"

Then there is the issue of understanding what we say when we speak and transcribing that to text. Our accent, the acoustics of the space, whether we have a cold or not, background noise, or other people talking at the same time all come into play to influence what sounds will reach the machine, which needs to then isolate the specific data it cares about and transform that into words. Once more, it's not just about a faithful transcription of the sounds into words. We *structure* things differently when we speak. We add "ums" and "ahs" and stop and start in strange ways that somehow all make sense to us but are not the same way we write.

As you can see, the challenges are considerable, and it is amazing that we now have readily available AI tools that allow us to recognize speech, transcribe that to text, understand its meaning, and even generate language. We haven't solved all the problems, but we've solved enough of them to make these tools viable for use in the development of AI-powered applications.

We briefly consider the implications of all this in the next section across speech recognition, natural language processing (NLP), translation, and natural language generation.

Speech Recognition

Speech recognition deals with our ability to transform the sounds that we produce when we speak to text. It is often also referred to as ASR, which stands for automatic speech recognition. Quite easily an entire field of study on its own, it combines a breathtaking set of technologies.

An ASR system starts by picking up the sound of our voice through a microphone. That signal gets cleaned and processed in the hope of isolating only those frequencies that represent a human voice. Those analogue continuous sound waves are then sampled and translated into what are referred to as *speech frames* (a couple of dozen milliseconds of sampled waveform information). Speech frames are then used to help us understand what *phonemes* the user has uttered. Phonemes are units of sound that combine to give us words and are used to differentiate between words—the linguist's equivalent to a grammatical syllable.[2] Linguists define the specific phonemes of each language

[2] For example, the word "Five" would be represented with three phonemes: "F-ay-v."

and how they combine into words; that knowledge is then used by ASR systems. This information is then further combined with a pronunciation model and a language model, nowadays largely based on deep learning, to produce the final text.

Speech recognition systems, especially after the huge enhancements that improved neural network algorithms introduced, provide an impressive amount of accuracy (all major technology companies report human level or better accuracy with error rates close to or below 5%). That does not mean, however, that we can assume that they will be able to tackle any situation with ease. The specific context needs to be taken into account, and a realistic investigation needs to happen into the viability of using speech recognition in order to solve a given problem. You probably already noticed how voice assistants are not that effective in crowded rooms with lots of other people speaking, whereas they perform much more reliably in a car where outside sounds are cut out.

The domain of discourse is also very important. Here is a very simple experiment you can run on your own to understand how it can affect speech recognition. Call up whatever voice assistant you have on your smartphone, be it Siri, Cortana, or the Google Assistant. First try telling them something that might be said in your work setting using domain-specific terminology, and then try an everyday phrase that is about dealing with more general life tasks. Look at the transcription of the text to see how accurate each got it.

I used the following work-related sentence:

"*The high-level objective for Task 1 is to produce a chatbot that is able to assist a user to search through multiple document repositories that are accessed through a federated search service.*"

This is a relatively friendly test. There are some domain specific keywords, but they are not too arcane. I am sure you, the reader, will have no difficulty with the individual words although you may have some questions about the overall meaning; for example, what exactly is a federated search service?

Google Assistant came back with:

"T*he high-level objectives for task wants to produce a chat but the table to sister user to search through multiple document repositories access with federated search service.*"

Siri gave me:

"*The high-level objective for task one is to produce a chalkboard that is able to sister user to search through multiple document repository other access through federated search service.*"

Those are admirable efforts, but not very usable.

However, if I try the following sentence:

"*Remind me to drop off the kids at school then go collect groceries, pass by the pharmacy, and then meet Julia for late breakfast*."

Siri gets it word for word correct and so does Google Assistant. I didn't even have to enunciate too carefully, something that I did do in the previous example.

Clearly, they work well for exactly what they were designed: to help us handle everyday life, rather than transcribe domain specific information. It is no surprise that one of the leading transcription software companies, Nuance, provides different software solutions for different industries such as legal, professional, and law enforcement. Each solution advertises the fact that it has been trained for that industry's specific vocabulary, precisely because that is a necessary precondition for effective operation in that industry.

In summary, although speech recognition has come a long way, it is important to keep a realistic view of where it can currently help, especially in an office setting. It can be extremely effective and less onerous to train if we want to use voice to issue straightforward commands or directions to a machine. In these cases, we are only uttering smaller phrases with a specific intent, such as "Open Microsoft Word" or "Call my HR contact." It becomes more challenging if we are trying to use it to transcribe complex phrases with domain specific (and especially acronym heavy) content.

Natural Language Processing

With speech recognition we go from sound to text. Once we do have text, how do we understand what we can do with it, though? This is where NLP comes into play. Let's look at some of the key stages to both understand what is possible and as a way to inspire ideas of how you can use it in your own work environment.

Analysis and Entity Extraction

The first stage is, typically, the syntactic analysis of the text we want to understand and something called entity extraction. Consider just a simple phrase such as:

"This is a book on the use of artificial intelligence in the office. It's published by Apress, part of Springer Nature."

To start with, we need to break up the text into its individual components; understand what constitutes punctuation and what does not, and how that affects the sentence structure.

Using Google's NLP demo[3] we get an analysis such as the one in Figure 5-1.

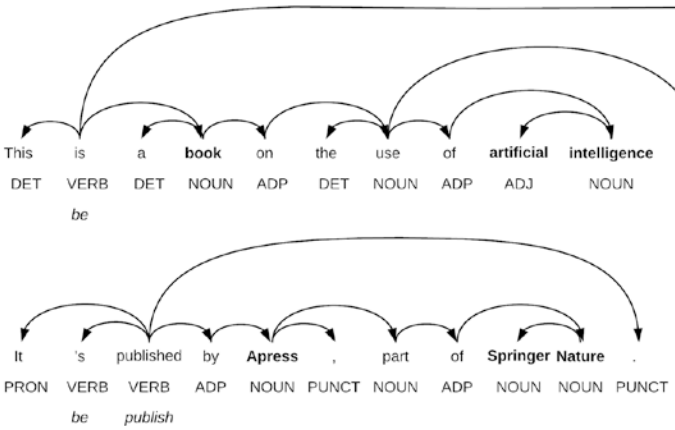

Figure 5-1. Syntax analysis of a phrase using Google's NLP API

You can see there is quite a bit going on. The NLP system has been able to successfully identify all the different words, including where we've used apostrophes such as "it's." It is also identifying nouns, verbs, punctuation, adjectives, and more.

Entity extraction is able to tell us that *book*, *Apress*, *Springer Nature*, and *artificial intelligence* are all salient entities in this piece of text and for some, such as "Springer Nature" and "Apress," it is able to say that they are organizations and provide links to their websites.

With just this information we can start thinking of a search powered by NLP that can be so much more effective than a "normal" search that only compares strings without any contextual information—a search that will, for example, be able to distinguish between when a specific organization is mentioned, such as Apple, instead of simply the fruit apple. Imagine being able to search through your document store and then filter against mentions of a specific company, product, or coworker names just the same way you filter against different brands on Amazon.com, without having had to painstakingly annotate those documents up front. The NLP system can do the heavy lifting for us.

[3] https://cloud.google.com/natural-language/#natural-language-api-demo.

Classification

What is the document about? Is it a sales report, meeting notes, or a pitch to win a new contract? Are the sentiments expressed within a document positive, negative, or neutral? Is the content sensitive or potentially offensive?

Classification, and in particular classification that is relevant to your specific needs, is one of the most frequent applications of NLP.

NLP tools have become particularly adept at this, and the good news is that it is already possible to train your own organization-specific classifiers with minimal specialized expertise. This is possible because you can base your classifier on existing language models (that have been prepared on much larger datasets) and specialize them with either rule-based classification or with data-driven approaches that work as a layer on top of the existing models.

Intent Extraction

When we are using language to communicate, especially when we are asking someone to do something for us, our words can be mapped to a specific intent. For example, if I say:

"Could you please open the window?"

The intent is quite clear. I am asking someone to open the window for me.

However, I could also say:

"It's hot; can you let some air come through?"

Although I didn't explicitly say "open the window," the intent is the same.

The job of intent extraction is to help us understand what action is conveyed in the words. As you can imagine, it is particularly important in conversational engines that power chatbots. They need to be able to map all the myriad ways we, as humans, can say something to a specific response or action. In addition, they need to do that while taking contextual information into consideration. Consider the following dialog.

> Human: "I'd like two pizzas, a Coke, and some garlic bread."
>
> Bot: "Thanks, what type of pizzas would you like?"
>
> Human: "A pepperoni pizza and a margherita. Oh, make that Coke a Sprite."

What to us is a very simple dialog is quite a challenge for a bot. It asked the user for the types of pizzas but it also got some information about a change in the order of the drinks. It needs to understand that the phrase the user uttered carried two intents, that the second intent was a reference to the previous phrase about drinks, and it's about changing the existing Coke to a Sprite!

Nowadays there is a wide range of tooling to help organizations develop applications that can handle such conversations, and problems like the preceding one can be solved in well-defined domains. The key is to clearly weigh where intent extraction and conversations would be most effective. It is a balancing act between the complexity of the NLP problem to be solved and the value the solution is going to generate.

Translation

We've probably all seen AI-powered translation at work. It is what makes possible those links on Twitter, LinkedIn, and Facebook that say "See Translation." It's what powers the translation feature of Google Chrome that translates an entire web page.

According to Google Research,[4] automated translation systems, under some circumstances, are approaching or even surpassing the translation quality that you would expect from human translators. It is important to take such claims with a healthy pinch of salt though. Those "circumstances" are important. If we are dealing with single words, short phrases, or web pages with small sections and not too complex concepts, automated translation can do an impressively effective job. The more sophisticated the concepts and the more layered the text, however, the less effective the translation.

A recent contest in South Korea pitching automated systems against professionals translating text from Korean to English and vice-versa concluded that about 90% of the automatically translated text was "grammatically awkward" and not comparable with what a skilled translator would produce.[5]

As such, the same limitations that we discussed so far apply here. Generic automated translation capabilities are impressive, but the more specific the domain the less efficient the translation model will be. If we are dealing with single words, simple commands, or small text, automated translation offers a viable avenue. For more complex scenarios, organizations need to evaluate the tools available and consider where they can invest in their own tooling if commercially available translators are not enough.

[4] https://ai.googleblog.com/search/label/Translate.
[5] www.koreatimes.co.kr/www/tech/2017/02/133_224449.html.

Natural Language Generation

The mirror image of natural language processing is the automated generation of new text.

We are far more likely to digest information and understand its implications if it is set in an appropriate narrative for us. We have all gone through that feeling of blanking out when presented with walls and walls of tabular data. Even with more pleasing graphs and charts, after a while it can feel like one blurs into the other. What we care about is the story that those tables and chart tell. Natural language generation (NLG) allows us to input structured, labeled data and get a natural language document that provides an appropriate narrative around that data as a result.

A particular strength of NLG is that it can produce multiple narratives from a single set of data, adapted or personalized to a specific situation. Take, for example, financial data. Analysts need to provide reports for all their different clients following the reporting of performance of a particular company or the release of data around a specific sector. The inputs, in this example, would be something like the annual company report and the portfolio situation of a specific client. An NLG system can then produce a narrative that describes what happened and how it affects a specific portfolio.

There are several levels of analysis that the NLG system performs to get to a final structured document. It needs to determine the relevant input data points that should be mentioned in the generated document. For example, did the company make a profit or a loss? What are the biggest expenditures? Where did sales mostly come from? The NLG system then manipulates what can be imagined as a very complex template that provides rules around the structure of the overall document and the structure of individual phrases. The end result is a document that is not only grammatically correct but structured in a way that is comfortable and natural for us to read.

Another example is weather reporting. From a single weather data set, a news organization can produce localized weather reports for all its affiliates without requiring a writer to go through the data and come up with appropriate narratives.

Within organizations, NLG is increasingly being used to provide the narrative around how the company is reporting in a more efficient and impactful way than charts and purely numerical reports. This can be particularly empowering for users who do not have the skills to do the necessary data analysis on their own.

Vision

Vision refers to a machine's ability to process visual data and interpret it appropriately. It can range from something as "simple" as scanning a bar code to identifying objects within a photograph.

The advancement in the interpretation of images, as we discussed in Chapter 3, is what opened the floodgates for more general applications of AI. Unlocking the ability to correctly interpret an image enables so many applications, from autonomous driving to the ability to better monitor and manage the growth of crops across large areas.

The toolsets to enable training of data-driven models (the overwhelming majority being deep learning models) is potentially the most evolved across AI capabilities. This is a combination of the incredible amount of work that has gone into machine vision[6] coupled with the suitability of deep learning architectures to handle raw image data.

There are powerful tools to label or annotate images that can then be used to train models and, unsurprisingly, there are several possibilities within a work environment.

- *Authentication and authorization*: Face recognition can be used to identify people and provide access to work office spaces. It is not without its challenges, though. It can offer a more seamless experience and, under the right conditions, a more reliable security environment. However, it comes with risks, as companies will need to store biometric data.[7]

[6] Just in terms of investment in autonomous driving, 2018 saw venture capitalists committing 4.2 billion dollars—the key technology developed there is real-time AI-powered machine vision: www.axios.com/autonomous-vehicles-technology-investment-7a6b40d3-c4d2-47dc-98e2-89f3120c6d40.html.

[7] In August 2019, for the first time, a large database of biometric data was found exposed on the open Web. It contained fingerprints and facial recognition data for millions of people and was managed by a company named Suprema. One of the biggest implications is that while one can change their password, if the digital equivalent of their fingerprint is stolen, there is no mechanism to replace it! www.forbes.com/sites/zakdoffman/2019/08/14/new-data-breach-has-exposed-millions-of-fingerprint-and-facial-recognition-records-report/#4cef3ee046c6.

- *Fraud detection*: In industries such as hospitality and retail machine vision can be used to detect when items are not properly processed at point of sales systems. They can monitor employees or clients as they are passing objects over barcode readers.[8]
- *Asset monitoring and management*. The analysis of images of physical assets can reveal where faults are close to occurring and optimize the maintenance of workspaces.
- *Digitization and categorization of analog documents*: We are a long way away from becoming entirely digital, and we have a swath of historical documentation that we still need to deal with. Machine vision can be applied both to categorize documents (e.g., identify receipts, sales reports, pay slips) and also digitize them so that the information within them is immediately accessible.[9]

Just as with NLP, there are powerful tools readily available to test out ideas with machine vision. A great example I've encountered, that tells the story of how accessible tooling has become, is of an intern building a fully digitized system for monitoring parking availability for the entire workforce over a single summer. They used the data coming from security cameras to figure out what parking spaces where available based on the movement of cars, exposed that information in the company intranet, and set up parking boards for everyone to see. This improved everyone's experience of coming to work and required, all said, minimal effort.

As with everything else, care needs to be taken to ensure that the capability you think you have developed can translate to wider deployment. Machine vision is notorious for providing false positives or completely missing the target. When dealing with life and death situations, such as autonomous driving, that is simply not acceptable. However, when the goal is to make an existing process more efficient (such as letting people know whether there is a parking space), some errors can be tolerated. Similarly, if we are using images to

[8] Beyond fraud detection, vision is also a core capability to automate the entire retail experience, as Amazon demonstrated with their automated grocery store, Amazon Go. The solution there relies heavily on cameras to track how clients interact with items in the store.
[9] A great example of on-demand-digitization is a new feature in Microsoft Excel whereby you can point with your smartphone camera at tabular data in a printed document and have that converted to digital spreadsheet data.

detect people in pictures in order to better classify a catalogue of media images, we are saving ourselves time and don't mind some errors. If we are using face detection technologies coupled with emotion recognition to determine the emotional state of a workforce,[10] we are definitely overstepping what the technology can usefully achieve and risk alienating users.

Custom Capabilities

Language and Vision are generic capabilities with wide applicability across different domains. Exploiting them appropriately across your organization can give you a significant advantage. There is space to innovate in how you use them and where you apply them, but it will become increasingly harder to complete with others on building better NLP or vision systems. The effort required will likely not justify the potential benefits for most companies. Ultimately, we can expect powerful NLP and vision capabilities to become the minimum standard necessary, rather than a competitive differentiator.

Instead, an area where there is possibility for your organization to differentiate and create more of a moat around your competitive advantage is in creating your own "custom" capabilities. These are ways you can represent, reason, and act in the world in a way that is specific to your organization because it is a result of models that you have devised and data that only you own. I like to think of these as your organizational superpowers. Just like hero superpowers, they are the things that separate you from the other superheroes. Some heroes can see better, while some can jump higher or pack a mightier punch. Those are their super capabilities. The question is: what is your organization's superpower when it comes to AI capabilities?

To develop a custom capability, you need to create the right circumstances. Just like the Hulk, Ironman, or Spiderman, you need to walk into a lab and mess around with the different ingredients to see what can come out of them.

For example, there may be something specific in the way you collect customer data that allows you to model and reason about the behavior of your clients in a way that others simply can't. You may have developed a culture and put in a place a process that means your team provides structured feedback in a consistent manner. This enables you to get a better overall understanding of team well-being and what needs to change, leading to a happier and better performing workforce.

[10] This is an application of vision that has been deployed in certain schools in China with the aim of classifying students based on six behavior categories, with the goal of identifying students who were not sufficiently immersed in study. Such applications of technology should rightly raise alarms: https://www.theglobeandmail.com/world/article-in-china-classroom-cameras-scan-student-faces-for-emotion-stoking/.

Perhaps, just like Superman coming from a different planet, you are entering a new market and can bring a perspective to it and new capabilities, in terms of how a process can be automated, that incumbents have simply not considered or are too comfortable to care about. The way fintech start-ups are disrupting traditional banks is a good example of this. They don't carry any baggage and are approaching the problem from a technology-first perspective in a way that the incumbents find hard to achieve.

The crucial element is to recognize that what you are looking to develop is a capability: a way to understand and reason about a specific aspect of the world. Starting from there you can then explore the techniques available and start combining them to get to a specific solution.

From Capabilities to Applications

AI capabilities are the ways that you can understand and manipulate your environment. Core capabilities such as Language and Vision offer a wide array of opportunities to organizations. There are easy to access tools making the barrier to entry low. The challenge lies in identifying the most fruitful ways of using them, cognizant of their limitations. Ultimately, these core capabilities will become part of everyone's toolkit. What is important is to grow the skills and experience to use them effectively now, in order to gain some first-mover advantage.

In addition, you can start thinking of what custom capabilities you can develop. These organizational "superpowers" can be exclusively yours because they depend solely on how you exploit the innovation capability of your people and your understanding of the world (your knowledge and your data). The more mature AI-powered applications become, the more important these custom capabilities become, as they are the ones that will provide true differentiation.

PART

II

The Applications of AI in the Workplace

CHAPTER

6

The Digital Workplace

Digital transformation.

Did the mere mention of this darling catch-all phrase of consultants generate a slight inner groan? I know it has that effect on me, and I *am* one of those consultants, working for consulting companies that invariably mention "digital transformation" on their web site's home page.

There is nothing inherently wrong with the phrase itself. Deep down we all know that. To transform processes through the effective use of digital technologies is a very sensible thing to do. It is something that every organization should always be doing. The reason the phrase produces dread is that it has been thrown around so much by earnest marketers of consulting services that it now carries a certain amount of baggage. Buzzwords like digital transformation, and its cousin, digital strategy, conjure up images of armies of consultants producing lengthy reports about what one ought to do to improve their workplace, with little practical advice about how to go about achieving that.

For the purposes of this chapter and what is coming next in the book, I ask you to put all that baggage aside. In this chapter, we are going to discuss the digital workplace, what that means, and how an understanding of the digital workplace forms the foundations of a strategy for the AI-powered workplace.

R. Ashri, *The AI-Powered Workplace*,
https://doi.org/10.1007/978-1-4842-5476-9_6

What Is the Digital Workplace?

Let's start by giving ourselves a working definition of what we are dealing with.

What is the digital workplace?

A simple way of defining it would be to say that the digital workplace is the list of all the digital tools that people use to get their job done. From the systems that run HR to e-mail systems, document sharing, reporting systems, laptops, phones, and meeting room systems, the list is long.

Having a list of everything involved is extremely useful and a task in and of itself, but is it enough? I don't think so. The problem with this definition is that it doesn't quite capture the real essence of what a digital workplace is. It is true that at some level it is a list of things. Software and hardware that come together to enable people in an organization to achieve goals. However, it says nothing about what that *means* to an organization and it offers no unifying view, which makes it harder to define an overall strategy. It would be the equivalent of defining the physical space that work takes place in as the list of furniture and rooms where that work happens.

The digital workplace is the digital environment through which work is done.

A more interesting way of thinking of the digital workplace is to say that it is the digital environment through which work is done. It complements the physical environment, and the two share the same overarching goal—namely, to facilitate work according to the vision, mission, objectives, core values, and principles of the organization. The latter part is fundamental in my view. Whatever process you are developing to solve a problem or tool you are choosing to support this process, you should ask yourself: is this in line with who we are as an organization? Does it reflect and support the correct vision and mission? Does it support our objectives and strategy to achieve those objectives?

Here's a useful exercise to get in the right mindset when thinking of the digital workplace: imagine yourself having to build a new physical space for your organization. What sort of questions would you ask yourself in that case?

Where will it be? The location (or lack of a single location if you choose to go distributed) will define how people interact with each other and with the outside world. What changes for your organization if you are in a science park close to Oxford, in the center of London close to startups and large tech companies, or in a Tuscan farmhouse holding video calls with your team spread throughout the world? In a digital environment, one analogous choice would be to consider the digital spaces that people use to communicate and collaborate (your e-mail systems, your project management tools, and your messaging infrastructure). Different types of digital locations will enable different possibilities.

How will it support the people working there? What is the experience people have when they step into your offices? What does it say about your organization and what sort of organization do you want it to be? People probably spend quite some time thinking about what the lobby of a building says about the company. They make huge investments in artwork or sophisticated fixtures. Typically, the older and larger an organization is the more impressive their lobby. However, it is far more likely that the first interaction with your organization will be via its web site. Is that as impressive as your lobby?

What sort of furniture and amenities would you have? Can people easily prepare a coffee? Are there areas that support serendipitous meetings and the exchange of ideas? Are there nicely designed meeting rooms that can facilitate creative work? Now, what if they have to hold a meeting online? Are they stuck with corporate tools that no external client can use, or is it as simple as sharing a link to run the meeting. Is there a way to keep notes for an online meeting, easily share screens, or record?

Finally, think of the underlying process of designing a physical space. You will contract an architectural firm, discuss a whole host of issues around what organization you are, what image you want to project, and what needs you have now and for a long time in the future. This is a big investment with a long-term plan. You then go through a number of ideas and finally start working on that new building. Have you ever had that in-depth planning process for your digital space? Can you pull a master plan out of a drawer that shows how all the pieces fit together to form a strategy? Do you have a process to shape how the choices around a digital environment get made?

I hope this illustrates the kind of thinking that should go into planning a digital workplace. Of course, the analogy is not a perfect match. The digital space in many different ways allows for much cheaper experimentation and exploration. You can't build and tear down numerous buildings, or keep moving the walls around. Well, you could, but it's not the most cost-effective way of doing things! You can, however, keep trying and evolving approaches in digital spaces. It is not without cost, but it is a cost that is far more easily justifiable.

Understanding Your Digital Workspace

In Chapter 3 we talked about what makes AI tick. We talked about the need to have a model of how the world works that will allow us to decide what capabilities we should apply in order to affect change in a way that is desirable for us.

When we are considering the application of AI to the workplace, the model of the world we are referring to is, naturally, the digital workplace. What are the components that make it tick? How do people, tools, processes, policies, and culture combine to make work happen?

It is worth reiterating here that when we talk about an understanding of the digital workspace for the purposes of applying AI techniques, we are not referring to some vague deeper meaning that may live in the head of a visionary CEO. We are instead referring to a very concrete digital representation of the workspace that can be manipulated through software for the purposes of automation. We need a model of the world that combines explicit knowledge together with knowledge we may uncover by analyzing data—a model that will enable us to predict future states that can be exploited through software to provide solutions that are useful for people.

In the next few pages we will look at the components that come together to form this model, and some of the dimensions that you should be considering in order to create your own model of your digital environment.

Before we start, here's a simple disclaimer. There is no single model that will just work. There are outlines that are more widely applicable, there is knowledge that we can transfer from one organization to the next, but ultimately you will need to explore your own space to understand what makes it tick and how to model it.

People and Teams

Unsurprisingly, it all starts with people and the teams that they form. What we mostly do within a workspace is interact with the rest of our team to jointly solve problems together. At least that is what we ought to be doing!

As such, having a clear model of the basic questions of who are our people, where are they, what are they doing, what can they do (skills), what do they know about (knowledge), how are they doing it, and even why are they doing it is essential. We can also take this one step further and say that we would also like to know the wider network that our people create when their connections to other organizations and people within those organizations are all mapped out.

There may well be a lot of pieces of data available, but they are often spread across different departments and teams and in different, often incompatible, pieces of software. The question to ask is whether there is a unifying way to represent people and teams within an organization. However, as we mentioned, the task can often seem daunting so we need a way in—a hook to get us started.

A useful way to start exploring how sophisticated your representation is of people and teams in the digital world within an organization is by picking clear use cases that you know would solve current problems, and then examining whether your current digital environment could support a solution.

Exploring People Data

For example, say you wanted to introduce an automated tool so that a group of people can set up a meeting simply by sending it a request. Something like: "Akeel, Barry, and Jasmine need to meet in a room within the next 2 days for 1 hour."

Is the information around people and teams available through your digital systems enough to make this happen? Think the problem through and make a quick mental checklist.

- Is it possible to check the diaries of everyone within a group?
- Do you have access to the meeting room calendars and their availability?
- Can you cross-reference people availability to meeting room availability?
- Could you include some geographical considerations and personal preferences around meetings, and then go ahead and book the meeting?

Few companies already do this. More have systems that could support it but are not doing so yet. And most wouldn't be able to even make use of such a tool because people's calendars are not accessible, meeting room timetables are not digitized, and there is no way of having a clear idea of who is where.

Here is another scenario. Your sales team just got in a touch with an exciting new client. They want to put together the best proposal possible and it would be amazing if someone internally already had a relationship with the client organization. So the question to ask would be "Hey, has anyone ever worked with client Y, knows someone from there, or has some sort of relationship with this potential client?" How would that play out now? How can you access the relationships and history of your own people? Would it have to be a company-wide e-mail? An announcement on a chat channel? Is there a less noisy way to ask the same question, facilitated by a system that more efficiently handles the communication exchange or access to relationship networks?

Here is a third example. Say you are facing a major challenge in the development of a new feature for a product. It requires specialist knowledge, and you are convinced that there must be someone who has expertise or access to expertise within the wider organization. Can you search for people based on their expertise? Is there a "people search" tool that would allow you to type in a person's skill and it can produce a list of people who have that skill?

Here is a final example. You are at the start of the year and would like to have a map of the availability of key people across dozens of teams throughout the year. Is there a model that would reliably predict the potential distribution of vacation time, sickness, or any other of the many events that means someone will not be available? If you would like to build such a model, do you have historical data that could be examined and analyzed to provide this information?

If the answer to all of the preceding scenarios is that it will probably be very hard to achieve within your own company, don't worry. That is the usual starting point. It's a wonderful opportunity. It means there is already low-hanging fruit that you can reach out and grab, and prove the power of automation to make the work environment a more helpful one for people and teams.

Processes

Following on from an understanding of who is part of the team is how the team gets things done. As I am sure many of you have experienced, sometimes all the information is there, and even the right tools are there, but there is a "rule" that things should be done in "a certain way." That becomes the blocker that stops something from being properly automated. Processes are what define these steps and the rules that dictate the steps.

Creating a map of all the relevant processes, what data is required at each stage, who is involved, and how information flows from one step to the other can seem like a laborious exercise to start with. Nevertheless, it is critical in allowing us to move with confidence in changing, improving, and automating them.

Mapping Processes

There is a wealth of systems to support business process mapping, from simple flow charts to elaborate techniques such as Six Sigma. As ever, there is no single solution. It is important to explore what will work best within your own organization. A simple way to start, which will also provide a definite lasting benefit if you have no structured information on processes already, is to create a company handbook.

A company handbook is a living document that is meant for anyone within the company to access to find out how things get done. From how to ask for holidays and time off to getting new equipment, training, or personal development, it can contain, in simple language, all the processes to achieve that. It does not require specialist knowledge to put together, and can be a collaborative effort across a team or organization as different people tackle different aspects of the handbook. With an explanation of the key issues in plain language, one can then identify what can potentially be improved or automated and then dig in and create more formal flow charts and process diagrams of those aspects.

Here's a final word on the mapping process. Try to avoid any references to specific software or tools in general when doing this. For example, saying something like "In order to request a vacation, one must submit a request via the HR vacation planning tool" is a representation of what happens but not of the real process behind it. What we are really interested in is what happens to that request after it has been submitted to the HR tool. Does it go to a line manager, the COO, the CEO? What should they do in response? That is the real process we are trying to map, so that we can then implement the process in any number of different ways and using different tools.

Tools

With an understanding of what is an appropriate model to represent people in your teams and organization, and a mapping out of processes, we can move to tooling.

What are the pieces of software that hold data and perform actions, and how are they connected? Once more, starting from specific use cases is typically more fruitful than starting with a blanket cataloguing of every single tool. The use cases will allow you to uncover specific information that is relevant to solving a real problem, and from there you can expand to map back to processes, people, and finally, the solution.

Demystifying Tooling

Tooling, I think, takes particular patience. More than anything else it can often be the trickiest mystery. When looking at a piece of software and trying to understand why it is the way it is, you are looking at a long history of decisions, power plays, limitations, and short-term solutions (hacks) that have produced what is currently in front of you.

Trying to untangle it all can often feel like too much effort to be worth the while. This is why it's important to have a clear understanding of the process you are actually trying to support (without reference to a specific tool) and how the data within the tool can support that process. You can then move with more confidence and swap out tools, knowing that the processes you care about will still be supported.

The Next Destination

The digital workplace is the digital environment that facilitates work. It is the sum of interactions between the people and teams that make up an organization, the processes that are followed, and the tools that are there to support these processes. Throughout, data flows from one place to the other, gets stored, manipulated, acted on, and transformed.

The digital environment should also be an expression of your culture and values. As cheesy as it sounds, it should be in harmony with your value statements. There is no point stating that you are an open organization, only for people to then find out that tools compartmentalize activity and do not allow people to freely collaborate.

A necessary precondition to introducing automation in the workplace is to understand your digital environment. A solid understanding of why things are the way they are allows you to move with confidence in introducing change.

In the next chapters we turn our attention to how AI and messaging platforms can be combined to give your digital environment an underlying operating system and user interface. We will see how messaging platforms can act as glue between disparate systems and as a window into the entire digital environment.

CHAPTER 7

AI Is the New UI

Artificial intelligence is often, and rightly so, brought up in the context of solving hard problems like discovering new genes, curing cancer, or enabling autonomous driving. There is, however, a far more mundane but as challenging set of problems that AI is already playing a key role in solving. AI is increasingly the magic sauce behind the software that manages our interactions with any computing device.

The aim of this chapter is to illustrate and motivate the links between AI and user interfaces (UIs), and demonstrate how AI-powered UIs are going to be important not just for consumer products but for the workplace as well. AI-powered interfaces will become a source of competitive advantage for organizations that use them correctly.

Moving beyond "point and click"

Since the widespread introduction of the graphical interface with the Macintosh computer in 1984, the predominant interaction paradigm with computers has been to point a cursor at something and click to select it.

Innovations along the way upgraded this basic experience, making it richer and smoother, but they haven't radically changed it. Yes, we can now use our fingers instead of a mouse. Yes, we can "pinch and zoom" with two fingers or "swipe" with more fingers. With some trackpads and smartphones we can even use pressure to cause different reactions. We went from tiny, underpowered processors with very little memory on grayscale screens to blazingly fast machines, virtually unlimited memory, and millions of colors. That's thirty-five odd years of improvements. Nevertheless, we are still pointing and clicking.

R. Ashri, *The AI-Powered Workplace*,
https://doi.org/10.1007/978-1-4842-5476-9_7

Don't get me wrong. All of these developments are amazing. The technology necessary to provide a smooth pinch and zoom experience is staggering. The fundamental paradigm, however, remains the same. You are manipulating objects on a screen (buttons, links, text, images) by using a device (a mouse, pen, trackpad, or your hands) to indicate to the machine what should happen to the object you are pointing at.

Interestingly, AI already plays a huge role in today's interfaces. A prime example is the virtual keyboard on your smartphone. It is constantly predicting the most likely letter you would have wanted to touch, which one you are likely to touch next, as well as what words and phrases you are trying to type overall. It is learning to adapt its predictions to your specific manner of touching keys and writing. If all of that was switched off, we would find it very hard to type any message on our phones. It is no exaggeration to say that the introduction of the iPhone was only made possible because it used enough AI techniques to make the UI possible.

These days, all the top-of-the-line smartphones have either facial recognition or fingerprint recognition. That is a feature that heavily depends on AI techniques to interpret the inputs it gets (your facial characteristics or fingerprint) to the ones it has stored in memory. The fact that they can do it in a seamless motion with practically no delay is nothing short of magic.

We are at a tipping point, however. It is time to move on from the point and click interface to something else. Additional AI technologies will allow us to take the next step, and there are three key underlying drivers.

First, as computing spreads to every aspect of our life and every device, the interface quite simply disappears or is not an immediate option. If you are multitasking, such as driving a car or preparing a meal in your kitchen, your hands are already occupied. Being able to speak to a computer is the only choice. If you are interacting with a device that is on your wrist, or embedded in your clothes or furniture, voice commands are the natural choice.

Second, it is about time we turned the tables on computers and the way we interact with them. So far, we have had to learn the "magic incantations": the sequences of clicks that will help us achieve our goal. Where in the endless layers of menus is the option we are looking for buried? Which of the various left-button, right-button, one-finger, two-finger, or three-finger with pressure click combinations should we evoke to make things happen? Why can't we simply tell computers what we want and have them do it? This has always been the vision, but now interface designers finally have tools to help them realize pragmatic versions of that vision.

The third driver centers on competition and how external forces make it inevitable for others to react. When a set of technologies reaches a tipping point and enables a new way of doing things, it provides a competitive advantage. This, in turn, causes competitors to look for ways to neutralize the advantage, which inevitably drives further technological innovation. The iPhone is a prime example of that. In that first presentation of the iPhone, Steve Jobs showed the state of the art in phones at the time: bulky, clunky, with physical keyboards. The iPhone changed all of that. In a few years the bulky and clunky phones were all gone. iPhone became the new standard by which smartphones were judged. Fast forward to 2019 and the iPhone is now competing to keep up with innovations that others are spearheading.

Now, imagine a support team that is able to provide a better customer experience because they can focus on the more complex cases while their automated virtual assistants, powered by conversational AI, are dealing with the simple and repeatable problems. As a result, all of their competitors will look to provide similar support interfaces for users, and the use of conversational AI becomes the minimum entry point.

As AI influences so many different aspects of what we do, these forces will cause change in many different ways. From a business perspective a great user experience cuts right to the heart of the efficiency issue. Imagine your sales team having to compete with a team that has ten times better and more efficient access to data, and the ability to create new visualizations and ask new questions of their data. While your team is trying to borrow the time of a software developer in order to write a new query to pull out a report, the other team can simply type or speak what they need in a conversational interface and have the results show up in the team messaging tool for everyone to share. We are past the point where a good user experience was a luxury to be added later, and we are quickly getting to the point where a good user experience will equate to active, smart interfaces that collaborate with users to solve problems. In other words, the interfaces of the future will be entirely dependent on AI.

In the rest of the chapter I will introduce some of the technologies that are enabling this change, and the interaction paradigms that they are making possible.

Conversational Interfaces

Our brains are hardwired for language. As toddlers we get to the point where we are learning new words every hour of our life, and often we just need to hear a word once and we can already start using it. Listening and conversing (whether through voice or gestures) is what humans do.

Now, compare that to navigating a web site or interacting with an app on the phone. That requires specific effort and training. We need to learn it explicitly and the rules keep changing on us. Different applications put buttons in different places, icons are different, etc. Conversations, however, remain simple: question, reply, response, repeat. From a human perspective, conversational interfaces as a way forward are a no-brainer. It's what we do all the time.

Conversational interfaces are digital interfaces where the main mode of interaction is a conversation—a repeating pattern of reply response.

A conversational interface can use purely written exchanges (e.g., within Facebook Messenger or via SMS); voice-based (e.g., with the Amazon Alexa service), or a hybrid (e.g., Siri or Cortana, where we use voice but receive replies in a combination of voice and text). Conversational interfaces can also provide rich replies that mix text with media, or simplify the conversation by giving us a set of options to choose the reply from.

In the next few pages I take a look at what is happening with voice and written conversational interfaces, and explain why I think we are at the start of a very significant change for both.

Voice

I remember as a teenager being completely enthralled by the technological achievements of the 1990s, arguing with my dad about voice recognition. A new software solution called Dragon NaturallySpeaking was making waves at the time. It was the first commercial software that claimed to effectively recognize continuous speech (at 100 words a minute!). It felt as though the days where we would only ever talk to computers were just around the corner. My dad was far more skeptical. While dictation software was impressive, he could see all the challenges of voice recognition in busy office environments with multiple accents from different cultural backgrounds. He could not see how voice could be the main interface with a computer anytime soon.

Dragon NaturallySpeaking was first introduced in 1997. I was convinced that by the time we got to 2000 it would be the dominant interaction paradigm for computing. I was a bit too optimistic. My dad was right. Natural language recognition was not anywhere near the required level of capability.

The history of voice-assisted products goes back even further. In 1962 IBM presented the first commercially minded solution at the Seattle World Fair. It was called Shoebox and could recognize 16 spoken words and perform mathematical functions. We've been trying to crack the voice challenge for at least the past 60 years!

Eventually, however, algorithms, data, and computing power advanced sufficiently. Now is the time to stand on the side of natural language and voice in the argument. There is still a lot of ground to cover but the ingredients are there. Anyone with a smartphone has a voice assistant in their pocket. Siri was integrated into Apple's iOS in October 2011. Amazon released Alexa in November 2014. Google Assistant launched in 2016, but the technology was gestating as Google Now since 2012. Every large tech company has a "voice" platform. IBM has Watson, Microsoft has Cortana, and Samsung has Bixby.

By early 2019 over 100 million products with Amazon Alexa built into them were sold.[1] This level of adoption is critical. Voice applications have many challenges to overcome before they become a stable part of our digital environment. The two main ones, however, are getting us into the habit of using voice to achieve tasks (even very simply ones) and doing this reliably in any number of different situations. Both challenges require broad adoption before we see results. Broad adoption means that we are going to be increasingly more accustomed to having them around, which will feed enough data back to developers in order to improve them so that they can perform reliably. This is why these devices are so cheap. The large tech companies know what they need to get the ball rolling, and the only way right now is to make it a no-brainer for us to purchase the devices. It is such a low price that we simply reason that, worst-case, they are a decent speaker or alarm clock!

Through this mass adoption strategy, the tipping point is getting increasingly closer. We finally have:

1. Natural language recognition technologies (both for going from voice to text and then understanding the meaning of that text) that are good enough and widely available enough to deal with well-delineated domains
2. Devices that can support conversation-driven interaction that are cheap enough and widely available enough
3. Development platforms that allow anyone to create conversational applications that can be released and reach a mass audience

This means that we will see an explosion in voice-driven applications as companies begin to explore the problem space and find those killer applications.

[1] www.theverge.com/2019/1/4/18168565/amazon-alexa-devices-how-many-sold-number-100-million-dave-limp.

Text

The same elements that are driving voice-based conversational applications are also driving text-based applications, but currently text has a few significant advantages. First, it is very easy to add text-based conversational interfaces to web sites and, second, messaging applications are the new kings and queens of the digital world.

The top four messaging applications have at least 4.1 billion monthly active users and on average we spend 12 minutes a day within messaging apps[2] (the fact that your most likely reaction to that number is that is seems low is further proof of how popular messaging apps are!).

Messaging applications are widely used in business as well of course, with Skype, Microsoft Teams, Slack, and many others used daily by millions of people.

The asynchronous but immediate nature of text-based interactions is particularly suitable for a very wide range of everyday tasks. From checking flight details, banking issues, to the latest updates from your kid's school, a text-based message is incredibly well suited. According to a Twillio survey[3] of users across the United States, UK, Germany, India, Japan, Singapore, and South Korea, 89% of users would like to be able to use messaging to communicate with businesses. For 18- to 44-year-olds messaging is preferred over e-mail or phone communications.

We are going to look much more closely at text-based conversational interfaces, the technologies behind them, and how they can be transformational for work in organizations in Chapters 8 and 9, so I will skip a more lengthy discussion here. The main takeaway, however, is that with messaging applications we are past the tipping point. It is where people are now and what they like using. Now, the question is if you are looking to take advantage of opportunities afforded.

Augmented Reality and Virtual Reality

No discussion of how AI will change the way we interact with machines could be complete without dealing with augmented reality (AR) and virtual reality (VR).

AR refers to interfaces that overlay digital information on our view of the real world, either through wearable devices like glasses or simply through the screen of our phone. As with so many technologies, the level of usage forms a continuum. You can go from adding just a couple of extra pieces of information

[2] https://www.businessinsider.com/messaging-apps-report-2018-4?IR=T.
[3] www.twilio.com/learn/commerce-communications/how-consumers-use-messaging.

to my real-world view, such as the name of a building or a small card with extra information, all the way to creating what are often called mixed reality (MR) environments where the digital layer is rich and can be manipulated.

VR, on the other hand, creates an entire new world and places us in it. Whereas AR or MR augments what we currently see, VR replaces the analog world with an entirely digital one. The user typically wears a head device that immerses them in the virtual world and holds interface devices in their hands to manage what is going, on or their gestures are "read" and interpreted through an external device.

Although these technologies are further away from hitting the mainstream than conversational interfaces, the inflection point is getting closer. Once more the magic sauce of better computing capabilities, better hardware, and the application of AI techniques in the form of machine vision, natural language, and much more will lead to solutions that have the potential to feel like a natural extension of what we currently do. Success, however, is by no means a foregone conclusion. Even when all the required technological elements are there, the use cases still need to be carefully considered.

For example, does anyone remember Google Glass? Released in 2012 to great fanfare, the device was hailed as the harbinger of the AR age. The wearer of the Google Glasses communicated with the device using natural language voice commands or by touching the side of the glasses, and the glasses were able to overlay relevant digital information just above your line of vision. All the ingredients where there: natural language, a wearable device, and tons of automation to make everything work smoothly. It was also a complete failure.

There are multiple reasons for why Google Glass failed, and this is not the place to perform an in-depth analysis. What is interesting from our perspective is that a lot of the problems had little to do with the technology itself. In other words, even if Google Glasses were the "perfect" device from a technical capability perspective, they still would have failed. They were expensive, created awkward social situations (e.g., concerns that people would be photographed through the glasses without being aware of it led to them being banned in various locations), and didn't solve an immediate pressing problem for people.

Unlike voice or text-based conversations that use an interface paradigm we are immediately familiar with, AR technologies add a new layer that we need to get used to. This means that unless it is done right, it becomes yet another interface to learn. If that interface offers sufficient benefits, people will invest the time to learn it even if it is not a great fit. If not, after the initial excitement, people will just give up. Indeed, calling Google Glass a complete failure is not fair. It has found uses in industrial settings where there are clear uses cases of helping skilled workers as they are completing tasks.

From a consumer perspective we have some strong examples of how AR can be very successful when it is used effectively. Pokémon Go from the gaming industry is perhaps the most well-known example. Pokémon Go gets users, equipped with smartphones, searching for and capturing digital Pokémon in the physical world. The game indicates to users where they need to go to find the Pokémon, thus giving it the ability to direct people to specific locations. The excitement of mixing real world treasure hunts with digital game play took the world by storm, and for a few months in 2016 it was impossible not to come across people either playing the game or discussing it. While that initial excitement has settled and we don't hear about Pokémon Go on news reports anymore, the game is still played by tens of millions of users and generates hundreds of millions of dollars in revenue.[4]

Practitioners learn through these successes and failures, and because the appeal of AR is clear, it will eventually break through and become part of the tooling that helps us get work done in an office. The first area of application, however, is more likely to be industrial rather than office-based work. The mix of costs/benefits in an industrial setting is far more obvious and the domains to operate in are well defined. A great example is from a company called UpSkill.io. They use AR glasses to help field technicians receive information from remote specialist support staff. The AR glasses create a two-way feed between the field technician manipulating a complex device such as a drilling machine and the specialist support person.[5] The back-end specialist can talk to the field technician, see exactly what the technician is seeing, and stream relevant information to the glasses. This allows the specialist technician to scale and support multiple field technicians, providing clear savings for the company.

Now, if AR is challenging because it introduces a new way of doing things, VR takes that challenge to a whole new level. VR technology needs to play the ultimate magic trick. It needs to make us think that we are in a completely new world but feel as though it is as natural as the physical world. For years the struggle was simply around packing enough computing into a portable unit so that you could actually wear the device and carry it around. A catalyzing moment was when Facebook purchased one of the most promising producers of VR headsets—Oculus VR—for two billion USD in 2014. The promise of the technology paired to the reach of Facebook convinced people we would all have VR sets in our living rooms in a short amount of time. Several years later the enthusiasm has settled but the technology has marched on. Oculus now has products that don't require any wires, and at a significantly lower price point.

[4] www.forbes.com/sites/insertcoin/2018/06/27/pokemon-go-is-more-popular-than-its-been-at-any-point-since-launch-in-2016/#5f67a02fcfd2.
[5] www.youtube.com/watch?v=tX6fWje-pRU&feature=youtu.be.

Ultimately, the promise of the technology is such that developments will continue. For our increasingly distributed offices, where large teams need to collaborate intensely on complex projects, tools that make that experience better are crucial. One of the VR holy grails is fixing the meeting room experience to makes those in the room and those calling in all feel as if they are in the same place. All the large technology companies and countless startups are working on VR/AR platforms that will put the tools in the hands of developers, to allow them to explore the space and find the user experience solutions and business models that will work. Some of the platforms to look out for are:

- Microsoft with its Hololens 2[6] platform is providing the raw ingredients to allow developers to build applications on top of it. It is currently predominantly marketed for use in industrial applications.
- MagicLeap, although a startup, has already built an amazing headset and platform to allow developers to build solutions on. They are focusing on entertainment experiences but also building the tools to create an AR experience for office work.
- Facebook, as we already mentioned, is heavily developing its Oculus platform.
- Apple has a mature AR development kit for the iPhone, and rumors abound about Apple AR glasses. Of course, no one can be sure until the official announcements come, but undoubtedly Apple with its existing AR platform will look to make the next move, which may well include some form of wearable device.
- Google has not given up on AR and VR technologies; it is simply taking its time to apply the learnings of the first attempt.

Overall, the promise of AR and VR is such that people simply cannot give up. What is interesting from an office work perspective is that in order to fully take advantage of these platforms once they are widely available, you will need automation to allow users to really interact with your organization's data and processes.

[6] www.microsoft.com/en-us/hololens.

Better User Experiences Are a Competitive Advantage

For a long time, software built for the office simply did not consider the user experience as an important feature. Enterprise software was serious software for serious people, and that meant that if you had to click through ten screens and memorize twenty shortcuts to get your job done, well that is just what you would have to do.

Thankfully, we are now not arguing that point anymore. Although a lot of software is still terrible, there is an understanding that easy-to-use software leads to better work due to less training for users, fewer things going wrong, and increased user satisfaction. Beautifully designed consumer electronics and positive user experiences with tools such as Instagram or Facebook also make workers demand better experiences at work as well.

The next phase is going to be about how we can introduce more automation into our software solutions and how we can further reduce the friction of interacting with them. This will become especially true as the problems we are trying to solve increase in complexity and the volume of work increases as well.

AI techniques combined with interface paradigms such as conversation, AR, and VR will play a key role here. No matter what the UI of the future is ultimately going to look like, it is clear that the organizations that are able to provide the smoothest interactions between their systems and their staff, clients, and partners will have a competitive advantage.

CHAPTER

8

Conversational Collaboration Platforms

I vividly remember the first time I saw a Telex[1] machine in action. It was the late 1980s on a visit to my dad's office. The machine noisily disrupted a quiet office by spurting out paper, furiously printing text as it went along, with someone hovering on top of it in anticipation of what the message was about to say. "It's a message from the office in Heidelberg," that person shouted. This weird machine, in an office in the UK, was woken up by a machine hundreds of miles away because someone was typing in Germany. Once the entire message made it through, it was read out loud, the team had an impromptu meeting to plan the response, and then that was sent back using the keyboard attached to the machine itself. As this was the late 1980s, it was probably one

[1] The Telex network dates back to the 1930s. It provided a network of teleprinters that could exchange written messages. It remained in use in businesses through most of the 1980s and was then eventually replaced by fax machines. If you have never seen one of them, imagine a networked dot-matrix printer attached to a typewriter! The operator would type in a message and send it and it got printed both on your side and the receiver side.

R. Ashri, *The AI-Powered Workplace*,
https://doi.org/10.1007/978-1-4842-5476-9_8

of the last few Telex machines in use. Nevertheless, it was my first experience of such a form of communication and I thought it was the coolest thing ever! It mixed the instant nature of voice calling without requiring synchronization between the participants as a voice call does. For an "Internetless" kid of the 80s, this was as close to magic as I could imagine.

Of course, fax, e-mail, and now modern instant messaging applications have made the noisy Telex technology redundant. Telex, however, was the technology that proved the value of (almost) instant written communication in offices around the world for over 50 years.

Nowadays, e-mail together with messaging-based collaboration applications like Skype, Microsoft Teams, Slack, and Facebook Workplace are an integral part of the digital work environment. In many ways they are as important as the building you work in or the desk you sit at. In fact, messaging applications are likely the only stable "environment" in a world where people are often on the move and remote working is on the increase.

In this chapter we will explore how the combination of messaging applications and conversational AI is going to lead to a new way of working and thinking about work. We will start with an overview of the state of messaging applications and how they have evolved to become much more than just a way to exchange messages. These new conversational collaboration platforms, enhanced with AI-powered applications, can have a lasting impact in how we get work done.

Conversational Collaboration Platforms

In Chapter 7 we talked about the rise of messaging applications both within organizations and as a means for organizations to communicate with users. We touched on how messaging applications are the fastest growing application type and how that creates an opportunity for business to talk to consumers (and their employees) in a whole new way.

In the same way that messaging applications on our phones such as WhatsApp or Telegram are much more than just a means to exchange messages between two users, messaging applications in the work environment have evolved and matured to support a range of activities. In fact, the evolution is such that calling them simply messaging applications or chat applications doesn't capture what they are really doing. It is far more appropriate to call them *conversational collaboration platforms*.

They are ***conversational*** because the primary means of interaction with other users on the platform (and quite often other applications) is through the exchange of messages. This gives us, the humans, the upper hand. We are very used to conversations, since it is how we already communicate and collaborate outside the digital domain. Conversations on these platforms can take different forms. From free-flowing conversations with colleagues in private 1-1 communication, to group discussions, to conversations with applications that will use a mix of natural language and more structured actions.

They are all about ***collaboration.*** The only reason we introduce this software into our organization is because we think it will make it easier for us to get things done. If they fail in that task, they've failed their goal. As we discussed in Chapter 3, we can consider to what extent they are passive in helping us achieve this goal or they are active participants. A simple messaging application that does nothing other than facilitate the exchange of messages is a passive participant. Increasingly, however, these tools are becoming active participants. Whether it is Slack, Microsoft Teams, or Facebook Workplace, their product development teams include AI experts that are working to make these tools more useful by introducing various forms of automation. Slack, for example, will highlight what it considers important messages, and it gives you the ability to sort messages you have missed "scientifically" in addition to more common choices like "newest" or "oldest." What they mean by "scientifically" is that a machine learning algorithm helped them order messages based on some measure of importance that they derived by monitoring your interactions in your Slack environment. The hope is that this will allow you to focus on the important things first, which, in turn, will facilitate collaboration with the entire team.

Finally, they are ***platforms*** because they offer a rich set of ways to add functionality to them. We can install applications that can redirect our e-mail to show up in a shared message board, help us better integrate with project management tools, or help us plan and coordinate meetings. We can also develop our own applications, unique to our organization, that can expose custom functionality to everyone, such as the ability to cause actions to happen in other applications.

As such, these conversational collaboration platforms can become the glue that connects all the different aspects of our organization (people, processes, and tools) and the interface through which we access them. They can become our organization's operating system.

Conversational collaboration platforms can become our organizational operating system, one that more closely resembles our organization and on top of which we can combine people, process, and tools to achieve our goals.

In the next section we will take a closer look at exactly what a conversational collaboration platform looks like in an office environment and why it is becoming such a popular tool. We will use Slack as our starting point, as it represents both a good example of the type of environment we are exploring and a simple one. We will also reference Facebook Workplace, since it expands the paradigm with some more "traditional" social networking ideas.

As a slight aside, Slack is also interesting as a company and startup. Their rise is exemplary of the significance of the space of conversational collaboration platforms. When Slack was originally announced, as a pivot from a different operational and business model, people had a lukewarm reaction. It was going after a space that most would describe as not particularly exciting. Why would you want to build yet another instant messaging application and compete with the likes of Skype, Google Talk, the old and trusted IRC (Internet Relay Chat) and the myriad of apps out there that allowed people to chat? What people failed to see, though, is that Slack was never trying to build "just" an instant messaging application. Slack was working on a conversational collaboration platform for the office.

Slack realized that while there were a lot of ways to solve the instant communication problem, there were few that were tackling the broader problem of collaboration in a way that teams could self-manage. This relentless focus on supporting work in a more general sense (and not just messaging) coupled to great marketing techniques, easy sign-up processes for teams, and effective design made Slack the darling of many office teams.

Slack was so successful that even teams that were already provided with an officially approved communications tool (usually Skype for Business) would install Slack "under the radar" because it represented a better, more fun way for them to collaborate. The company grew in just a few years to revenues of $401 million in 2019 and a market value of $15 billion. In the meantime, Microsoft has responded with Microsoft for Teams (the "Slack" of the Office Suite) and as of 2019 was claiming more users than Slack, since they could exploit their enormous reach into all types of organizations. Facebook, on the other hand, has a quiet (in media terms) but huge (in numbers terms) success story on its hands with Facebook Workplace, which is exactly what the name betrays. Its promise is that it provides all the familiarity and power of the most popular social network and messaging infrastructure exclusively for your business. Companies like Walmart, Spotify, Starbucks, and Heineken are using it as the common platform for their entire workforce.

Core Features

Conversational collaboration platforms need to tackle at least six key areas to act as a strong starting point for the organization's digital OS. They need to

- Provide a way for us to find and interact with everyone else in the organization.
- Indicate what the team is doing and who is available.
- Support 1-1 as well as group conversations in a variety of configurations based on project needs.
- Support the exchange and display of documents and links.
- Allow search across both conversations and documents.
- Provide the means to integrate with the outside world (i.e., the rest of the organization's digital estate as well as outside partners and service providers).

People

The first step is, of course, to gather the people on the platform and be able to discover them in useful ways. In this context, it is informative to consider the differences between Slack and Facebook Workplace. Slack starts out as a messaging platform and grows into a conversational collaboration platform. Facebook Workplace starts out as a social network and grows into a conversational collaboration platform. The different starting points are evident in how they handle people information.

On Slack, each member has a profile with some very basic information (photo, e-mail, a "what I do" field). Users can add additional fields, and Slack offers both "built-in" fields such as a link to your LinkedIn profile or the ability to create custom fields. Workplace, on the other hand, betrays its origins by starting out with a much richer profile including fields for things such as skills and departmental information.

In either case, what is significant is the ability to create profiles with *structured* information so that software can take advantage of that information in automation processes. Going back to the AI techniques mentioned in Chapter 4, this structured information can, for example, power a broader knowledge graph of people and their skills. Depending on your specific needs, you can add fields that describe information that your organization cares about. Through the platform, people can self-manage their part of the knowledge graph, which can then be aggregated and reasoned over.

You could either contain all relevant information within the conversational collaboration platform or integrate it with a more powerful external tool that pulls information from the platform. This information can then be used to power searches to find people in the organization, based on a variety of parameters such as skills, departments, geographical location, and so on.

Presence

The green dot next to a person's name, indicating whether they are logged into their messaging application or not, is the digital equivalent of glancing across the office to see if Sarah is at her desk. However, unlike the analog world, we can make that green dot more interesting by posting a specific status update next to it such as: "please don't disturb," "happy to chat," "trying to focus," "having lunch," etc. We are providing context that in the analog world could be hard to replicate. This is useful information that both colleagues and automated programs can exploit to determine how best and when best to interact with someone. A simple example of utilizing this is integrating your calendar with your status information. Now people can know that even though you are at your desk you are actually in an online meeting, or you are working through time you blocked on your calendar for a specific task.

These types of integrations are the building blocks for increasingly more intelligent and interesting automation. Automation needs a clear as possible model of the world it is trying to automate. Presence information is a piece of that puzzle.

In addition, it is not just information that is useful to describe the current state of things. Presence information *over time* also gives valuable information about the ebb and flow of work in an organization. When do people log on, how long do they stay logged on, how do other events (company meetings, conferences, etc.) influence the availability of people?

Conversations

Conversations are, of course, at the heart of any messaging application. The key aspect here is the application's ability to support the richness of different types of conversations that may be required. In the same way that in an analog office we congregate in different places for different reasons, the digital space must allow it. You *can* have digital watercoolers, kitchenettes, private meeting rooms, and town hall assemblies.

There are three dimensions to consider: the structure of groups, the type of conversations, and control of access to groups.

First, in what configurations of groups (or channels) are we able to converse? We need to be able to have 1-1 conversations, ad hoc group conversations as well as longer-term topic-based conversations. The different configurations will end up being a reflection of how your organization operates. At times it may be a truer reflection of the real groupings than what your official organizational chart reveals.

Second, what type of conversation should be taking place? One example is the difference between a free-for-all open-ended conversation and a more Facebook-like interface with a main long-form post and comments below that. In Slack that is partly replicated with threads, where a user can reply directly to a specific message within a channel, making that top message the equivalent of a post in Facebook. There are also applications that attempt to marshal conversations so that clear outputs can be highlighted and specific decisions are pulled out of the flow of input.

Finally, from a security perspective you need to consider

- Who is able to join or leave groups?
- When a new member joins a group, do they have access to the historical information in that group or not?
- Who can create private groups and what are the rules that govern these behaviors? If an organization is striving for transparency but people realize that there are numerous private groups that they cannot join, what does that say about the organization?
- In general, are conversations to be considered truly private (even 1-1 conversations) or is the organization planning to access that information. In simple terms, if I am chatting away with a colleague in a private conversation on the organizational conversational platform, who might eventually view that information?

Different situations and different types of groups will call for different approaches. What works best is something that each team needs to explore and experiment with. The one fundamental point here is that these issues should not be considered minor details or secondary. In just the same way that one would not allow physical spaces to be arbitrarily designed, your digital spaces need the same care and attention. As we discussed in Chapter 6, everything feeds into what defines you as a group. If you are not intentional about what culture is being developed, you may end up with a culture and processes that are not only unsuitable but also very hard to change.

Next, I outline some examples of different types of groups, based on configurations I've seen across multiple environments, to give you a more concrete view of what different setups can look like.

- *Community of practice groups*: These groups support people working in the same field across an organization (e.g., all front-end developers, all project managers, all digital marketing people) to share information and chat about the way they do their work.

- *Project groups*: Project groups provide a space for a cross-functional team working on a project to gather and focus on just that work. They last as long as the project requires them, and the participants can change as people join and leave the project.
- *Announcement groups*: These groups are meant for team-wide announcements and minimal other types of interaction. It makes it easy for people to track these announcements without them getting lost in the chat flow of busier channels, and also makes it easy for people who were not around to catch up. Depending on the platform you are using, these groups can also be made read-only so that people are not tempted to discuss the announcements in that same context, making information clearer to follow.
- *Fun and hobby groups*: Are there a lot of ardent cat lovers in your team, or perhaps pub quiz aficionados or sport nuts? Create separate groups for them and let them enjoy the chit-chat around their favorite topics, and build bonds across communities of practice, project teams, and organizational divisions by discussing their favorite topics.
- *Shared groups*: This is a Slack-specific feature, but one that I believe is very exciting. Within Slack you can create a group (or channel, as Slack specifically calls it) that is shared across different Slack organizations. What this means is that people from company A and company B can access a shared group to collaborate, with Slack handling all the upfront authentication and authorization issues. This is a great way to create shared spaces for you and your partners to come together and collaborate.

There are several other options and variations to the aforenamed groupings, but I hope you can see how incredibly useful it can be to organize a digital space like this. As mentioned earlier, the key is to be intentional and make sure that the team considers (and keeps reconsidering) what the norms and regulations are that should govern the creation of these groups. Let me repeat it once more: in the same way that in a physical space you wouldn't want people moving desks ad hoc or knocking down walls, you need to ensure that it doesn't happen in the digital space.

Finally, as with presence information, the understanding of what groups people belong to and how the shifts evolve over time can be incredibly powerful. A lot of effort is spent in understanding the network dynamics within organizations.

Collaboration tools like Slack can provide the strongest signal yet of what is actually happening in the clearest way for organizations to exploit. You can directly map who is talking to whom, what teams people belong to, and you can even explore the ways people are collaborating. Are they being constructive or critical? How does their behavior change across different teams and contexts?

Document Exchange

Another aspect of collaboration is exchanging relevant documents. Tools like Slack make that extremely easy, since they allow you to simply drop files in the flow of a conversation with an associated message. Now, one could argue that this is a bit too easy, and I can attest to how confusing it can get at times. Imagine working on a document and discussing it with a team with new versions of that document constantly being dropped into the flow of a conversation. It becomes tiring to keep track of what you should be focusing on. The reasoning for sharing documents this way is that documents are part of the conversation and work, so you can keep everything in context. The promise of conversational collaboration platforms is that search and integrations will then be able to surface those documents once more.

Indeed, some progress is being made. For example, Slack has a very useful integration with Google Documents. When you share a Google Document you are not sharing the actual document but a link to the current version of the document, which at least removes the question mark of whether you are working on the latest version. The catch, however, is that when you then go to the document, you discover that there is a whole other conversation taking place within the document itself! Google Documents allows users to exchange comments and chat within Google Documents. So now you have two conversations going on. The one in Slack, where perhaps people who are not directly working on the document had some interesting things to say and then one in Google Documents. That's clearly not an ideal situation.

Overall, I have yet to see an application that has really cracked how to properly deal with this. We are dealing with separate collaboration tools and potentially different modes of collaboration. At times we share links to documents while at other times we share the document itself. These documents and links become part of a conversation, but search is not always as effective in finding what we really need; and while integrations can help, they can also add complexity.

Norms can once more help here, since teams can clarify how they are supposed to work. However, this is definitely an area where more work is required to give us the type of collaboration space we really need. It is informative to look at tools like Dropbox that now also offer a collaboration environment, but one that is much more document-centric (since that is their heritage). Ultimately, we should see conversational collaboration platforms evolve to encapsulate a much more document-friendly toolset.

Search

Search across people, conversations, and documents is what can tie it all together, and it's a crucial function for any conversational collaboration platform. Knowing that you can rely on powerful search is liberating because it encourages higher levels of sharing and engagement, since people are confident that they will be able to find that information again.

Current search capabilities are impressive. For example, on Slack, you can create a search that is looking for a file that was shared in the "2019 Annual Conference Planning Channel" by Tim or Mariola, within a certain date range, and has the keyword "Sponsors" associated with it. The speed of response is near instant and the quality of results is high.

Nevertheless, while these types of searches are impressively powerful, they also require consideration and focus from the user to construct. It is too easy for users to get lost in all the choices or forget what combinations are even possible. Overall, while search is already very useful, there is a long way to go still and (unsurprisingly) it is one of the main areas where the further application of artificial intelligence techniques will allow us to make progress.

Ultimately, I would hope to see search that is much more conversational and combines access to a number of different document sources from a single location. It should be more conversational, so we can search in the same way that we think rather than have to adapt our thinking to the filters and options that the tool provides. We should be able to use natural language to describe what we need, and rely on a combination of conversational applications to help guide us through the possible options. It also needs to take into consideration the fact that information lies across many different places, from our e-mails to our document stores and our conversational collaboration platform. We need paths from the platform to the outside world so that we can connect it all together.

Integrations

So far, we've talked about collaboration environments as the place where a team discusses issues and make decisions. Integrations with the outside world means that these same collaboration environments can also become the place within which actions are performed and change is affected.

Let us take one simple task and consider how different it can look with and without integration. The objective is to know when someone has asked for some vacation time and approve that request, reject it, or ask for more information before making a decision.

Without integration in your collaboration space, you are hopefully going to get that notification via e-mail or the eager colleague, who knows you only check e-mail a few times a day, will message you to say that they've put in a request. After all they are anxious to get an answer so that they can continue planning their vacation! Once you do see the request, you will likely head to your browser and log in to the HR software (hopefully it is single sign-on; otherwise you will need to retrieve the password and do a 2-factor authentication). Finally logged in, you start clicking around a bit to figure out how to get it approved. Eventually you get there. You then realize that you are going to have to check in with a couple of people. Other e-mail chains are fired off and you have the mental load of ensuring you check in time, connect all the pieces together, and eventually make a decision. Now you have to remember what it was you were actually trying to do before this entire process started and get back to that.

With integration you will get a message in your conversational collaboration environment. You immediately know that it is about a vacation request since it is coming for VacationBot. The message contains all the information you need, and you can click "Approve," "Reject," or "Discuss." Clicking "Discuss" creates a private channel and drops both you and the person requesting time off into it. There you can quickly ask a couple of planning questions and perhaps invite anyone else who needs to have a say. The entire conversation, with the relevant participants, is in one place. Eventually satisfied, you click "Approve" and the channel the discussion happened in and the bot go away.

Integrations can be incredibly powerful and can transform the collaboration tool into a full-blown operating system for how you run your organization. All the main messaging collaboration environments provide dedicated app stores where you can find a very wide range of applications and integrations.

These types of applications are what I call conversational applications and they are the focus of Chapter 9. There is just one other type of conversational platform that I would like to discuss before though.

Custom Conversational Platforms

So far, we've talked about conversational platforms with the underlying assumption that we were dealing with software that came from large technology companies (such as Slack, Microsoft, or Facebook). Before we move on, it's important to note, however, that conversational platforms can come in a variety of sizes and shapes. They can range from platforms that focus on specific verticals (e.g., finance, health) to fully customizable conversational platforms developed for the unique needs of a particular organization and application.

An industry requiring a different approach is the finance industry. Traders, just like any other profession, can benefit from instant messaging and collaboration but require a degree of security and conformance to regulations that goes beyond what most industries require. This has created the conditions for a thriving vertical of messaging applications that specifically cater to the finance industry. Tools such as Symphony,[2] offer the required level of security coupled to features such as appropriate audit paths, trusted identify management across organizations, and data ownership.

Similarly, given a large enough problem, it might also make sense to build a completely custom conversational collaboration platform from the ground up. As part of work that my team has been doing for the accountancy firm BDO, we built a custom conversational collaboration platform precisely because it is only through a customized tool that we could tackle the problem.

CONVERSATIONAL TOOLS FOR AUDITS

The problem that BDO was looking to address was how to streamline their audit process. Expert auditors from BDO interact with the client over a long period of time, and data is gathered from a number of different sources in order to complete the audit. While there was some tooling available to help with the process, such as a shared document store, ultimately it involved a lot of ad hoc back and forth, e-mails, and placed considerable demands on the entire team to coordinate so as to avoid confusion.

After some initial experimentation we decided that the best course of action would be to build a dedicated tool that would allow audits to be designed and completed. The tool is a conversational collaboration platform in that it allows the auditors and the auditees to exchange messages during the audit process, with specific groups set up to handle the different aspects of the audit. All the audit information is collected in appropriate ways through the application (with custom forms and fields). In addition, an automated tool—a conversational application—acts as the coordinator, providing proactive and reactive support to both auditors and auditees as they are going through the process. It can react to requests for help (e.g., "how should I be handling VAT in this region?") but it can also proactively offer suggestions based on the type of data that is entered. The overall platform also interfaces with other BDO systems so as to streamline the transfer of data.

[2] https://symphony.com

The end results were

- A much more user-friendly audit process, which leads to higher quality audits
- A much better understanding of audit data, as it is collected in a structured way
- Complete control and security over audit data through a wholly-owned platform
- The opportunity for BDO to capitalize on the advantage of having lowered the overall effort required to complete an audit while increasing the reliability of the audit itself

Building a custom chat application from the ground up is not an unrealistic proposition, even for smaller organizations. If you feel that there are enough benefits to be gained from completely controlling information flows and behavior on the platform, then it is possible to combine a number of different existing technologies in order to create an environment that is within your control and entirely owned by your organization.

Your Organizational OS

In this chapter we've dissected what it means to have a conversational collaboration platform and how it can act as an operating system layer for your organization. These platforms can delineate the space within which your digital environment can grow.

With features such as people management, presence information, group management, and so on, these platforms can allow you to structure how work is done. As with anything, it is important to ensure that you proceed with specific purpose, so that the resulting processes are a reflection of the type of team and culture you are looking to have.

By integrating these collaboration platforms into other aspects of your organization, you convert them into the interface through which potentially every other aspect of work can be accessed.

These integrations, or conversational applications, can be a key differentiator for your organization and through automation they can provide significant gains. In the next chapter we examine in more detail what conversational applications are and how you can use them within conversational collaboration platforms.

CHAPTER

9

Conversational Applications

In 2016, Microsoft CEO Satya Nadella declared that "Bots are the new apps."[1] The world at the time was riding a wave of optimism around what chatbots would be able to do. It was clear that messaging applications were growing faster than social media, and users were suffering from app fatigue.[2]

Chatbots looked like the ideal way for companies to position themselves where users were (i.e., within messaging applications). In addition, since adding a chatbot was as simple as adding any other contact to your messaging app, it meant that you didn't have to download and install something separate.

Others took this optimism further still, declaring that *web sites* were done for as well. Why have a web site when you can directly talk to a brand on Facebook Messenger? Chatbots were going to take over the world and nothing could stop them.

[1] https://eu.usatoday.com/story/tech/news/2016/03/30/microsof-ceo-nadella-bots-new-apps/82431672/.

[2] "App fatigue" is a term used to describe the reluctance of users to install yet another application on their smartphone.

R. Ashri, *The AI-Powered Workplace*,
https://doi.org/10.1007/978-1-4842-5476-9_9

By 2018, people were singing a different tune. The impending takeover of chatbots did not quite go that way. Web sites were doing just fine, and so were applications on our phones. People had interacted enough with chatbots to realize that they were not that smart after all. Actually, quite a lot of chatbots where ineffective and would trip up at the slightest problem. What happened?

Well, chatbot technology, like any other technology, went through a natural cycle of maturity. The initial enthusiasm, necessary for anything to get any traction, met the harsh problems of the real world. People tried out different approaches because there were no established best practices yet. Just like those web sites of the late 1990s and early 2000s, we added the equivalent of blinking cursors and rotating GIFs. Everyone did things slightly differently, and users were trying to figure out what on earth was going on. Inevitably, people got disappointed and some dropped away.

At the same time, however, practitioners learned from their mistakes. Conferences and meetups took place and lessons were shared. Some made it through the dark times and the process starts over. This time, however, experience and redimensioned expectations can lead to more practical applications.

By 2019, chatbots were no longer considered the savior of all things digital. Far more sensibly, they were considered another useful tool to apply with due consideration. My experience with chatbots tracked this trajectory. I recall sitting down with Tim Deeson in 2016, my now cofounder at GreenShoot Labs, and discussing what technologies would be interesting to explore. Chatbots and conversational AI were at the top of my list. By 2017, we decided that it was time to do something practical. We started building some experimental chatbots to see what the technology was capable of and discussed the possibilities with clients.

What we learned over the next year is that although there was a lot of activity around consumer-facing chatbots, there was also a very interesting but less explored opportunity with enterprise chatbots: chatbots specifically built to help people within organizations to get work done. We went ahead and built a product for Slack, a bot called TeamChecklist that allowed users to store useful checklists and manage them as a team. Through that process we discovered just how many different applications of chatbots there were, but we also clarified our ideas about how we should think about the space.

Ultimately, the more interesting product we built was not TeamChecklist but the tool we used to build TeamChecklist! OpenDialog.ai is an open source conversation management platform. It expresses a very specific way of both designing and building chatbots. GreenShoot Labs eventually pivoted into a conversational AI consultancy that helps clients figure out what solutions are useful for their specific situation and, where required, uses OpenDialog.ai to implement those solutions.

In this chapter we are going to present some of the ideas that underpin our understanding of applications that use

1. Conversations as their main interaction paradigm
2. Conversational AI to provide users with a more natural way of exchanging messages

As with all the other chapters, I feel that clarity around concepts and the relationships between them is crucial in order to be able to plot an effective strategy.

From Chatbots to Conversational Applications

One of the problems of chatbots is that they suffer from a lack of a clear definition. At the core of this "identity" issue is the conflation of issues surrounding the use of artificial intelligence with a mode of interaction between humans and machine. When people try out something that responds "like a human," the almost knee-jerk reaction is to test the limits of its capability to respond. Admit it. We've all messed around with Siri or Google Assistant: asked them questions they had no business being able to answer and giggled at their responses. This overlaying of different issues (how intelligent is it? what can I do with it? how does it interact with me?) can be distracting though. It is for these reasons that I prefer to call what we are building "conversational applications." The use of the term application reminds people that we are trying to address a specific need, and it just so happens that the predominant mode of interaction is via a conversation.

A conversational application is one that is meant to operate within a conversational environment, such as Slack, where the predominant mode of interaction is the exchange of messages.

Using the term conversational application makes it clearer that we are attempting to build a tool to complete specific tasks, as opposed to a more consumer-facing chatbot that might be just as equally employed in simply entertaining the user with chit-chat. Broadly, it is the same distinction I would draw between more general web sites (e.g., a news web site or a social media web site) where the goals are more open-ended and more specific web applications like Xero for accounting or Google Docs for online document collaboration.

Structure of Conversational Applications

What makes up a conversational application? While we don't want to delve into the specifics of implementing applications, it is useful to understand the environment within which we are operating. This will help you visualize how you could integrate conversational applications (both off-the-shelf and custom ones) into your own conversational collaboration platforms.

The diagram in Figure 9-1 provides an overview of the main components, and we will look at each in turn.

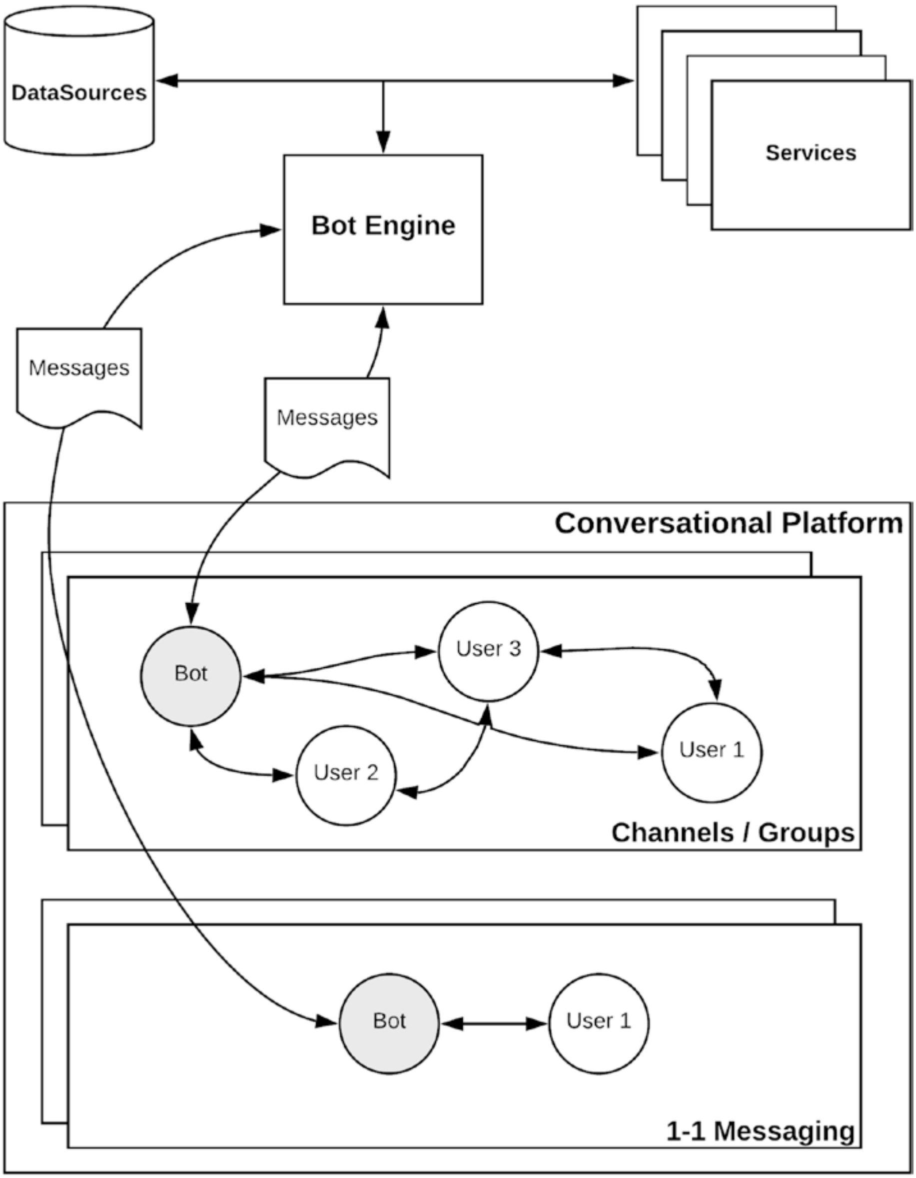

Figure 9-1. Elements of a conversational application

Central to it all is, of course, the bot itself. The bot is the identity that our application assumes in order to interact with users in a conversational platform. Just like every other user, it has a name, an avatar to represent it, and can provide presence information. It is, for all intents and purposes, another participant in our platform. Through the bot engine, the bot can interact with data sources and services in order to provide relevant information and carry out tasks.

Users can interact with bots either within groups (where multiple people can exchange messages and collaborate) or through direct 1-1 chats (where just a single user and the bot participate). If it is part of a group it can, depending on the granularity of settings for applications within a platform, "listen in" on all conversations happening in that channel just like any other member. This can be useful if the bot is meant to proactively participate in discussions, but if the bot is only supposed to provide information and reply when it is directly engaged, it is best to look for ways to limit the flow of information to the bot.

■ Keep in mind the security implications of allowing a bot to participate in channels where sensitive information may be shared. If a bot can listen in on everything, this means that necessarily all the information is flowing back to the bot engine. If that bot engine is a component that is internal to your organization, that may be OK or at least easier to manage. But what if that bot engine is actually an SaaS product? Who is the provider and what guarantees do you have that they will treat your data appropriately? You need to carefully consider what information you are potentially sharing with them and what the terms and conditions are that govern this.

■ Allowing a bot onto your conversational collaboration platform and giving it permission to listen in on all conversations is the same as allowing a stranger to walk into your office and listen in.

Having a bot in a channel can also be useful for a group of people to see the results of an interaction with a bot. For example, when someone asks for the latest sales numbers from a bot, everyone can see that report. Conversely, you need to consider whether that is too much noise for a team or simply not relevant. Often the route that conversational applications take is for notifications to arrive in a channel, while the execution of operations, actions, or the request for further information happens with 1-1 messages.[3]

Messages that the bot receives make it back to some sort of "bot engine" where reasoning will take place about how the bot should reply. Depending on the type of message, this is where we would be employing natural language processing to extract specific meaning from a user phrase and map that specific meaning to a response or action.

---

[3] There is some more nuance here, in that bots can post messages in groups that only the user of the bot sees. For example, if you ask a bot within a group to perform an activity and share results with the team and that activity fails, the bot can show just you the error instead of adding noise for everyone in a group.

However, not every message will need to be treated as a natural language phrase. The bot can present the user with a variety of standard interface elements such as buttons, dropdown form elements, checkboxes, and links. Just like when we interact with forms on any application, interactions with these elements (e.g., the user clicking on a button called "Approve") provide our bot with unambiguous information that it can act on. Consider Figure 9-2 as an example. In the open-ended interaction scenario, the user is asked whether they want to renew their membership, and the application is expecting the user to type a response in the same way they would reply to a friend. Indeed, the user decides to answer with "Sure, why not!" Will the NLP system helping us identify responses be able to map that to a yes? It might manage that, but imagine all the different ways a human can answer. They could say: "No, no, yeah, go for it." In the structured interaction the user is instead presented with actual options that our system will understand: "Yes," "No," "Remind me in a week." That leaves much less space for things to go wrong.

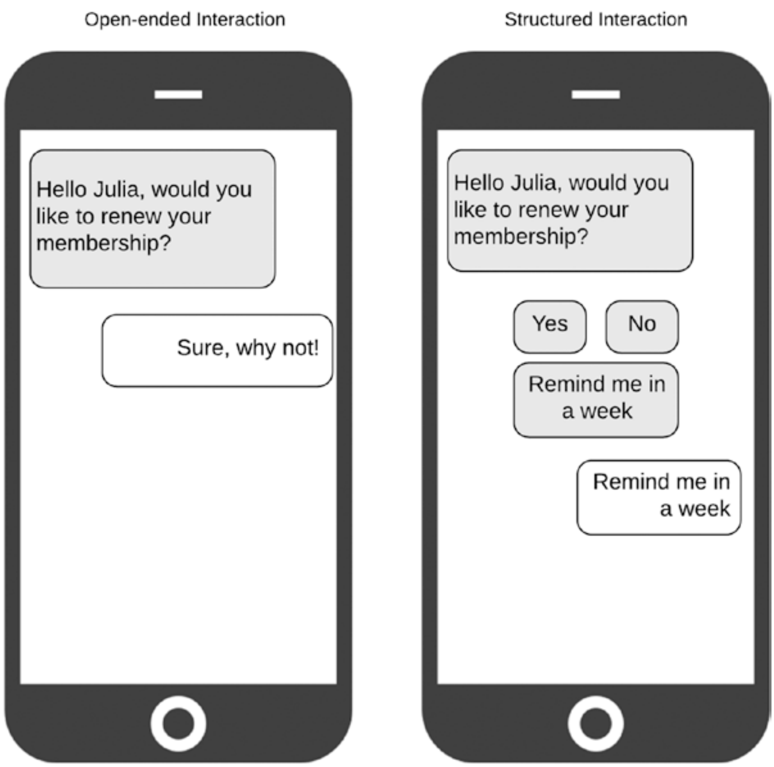

Figure 9-2. Open-ended vs. structured interactions

I have often found myself debating with people about the *authenticity* of a chat experience on the premise that structured exchanges, where people can press a button or select an option from a dropdown menu are considered *inauthentic*. "That's not a proper chatbot; it's just a glorified form experience," the argument goes. While I can understand where that thinking is coming from, the first consideration must be the ease through which a user can flow through the interactions. If you can clarify options for a user with a couple of buttons, then that has to be the right thing to do. While we are taking advantage of the conversational nature of the interaction, we should not lose sight of the fact that we are in a richer digital environment and we have access to a wide range of options.

In general, it may be useful to consider interactions with conversational applications as potentially lying across a broad spectrum of possibilities, as Figure 9-3 illustrates.

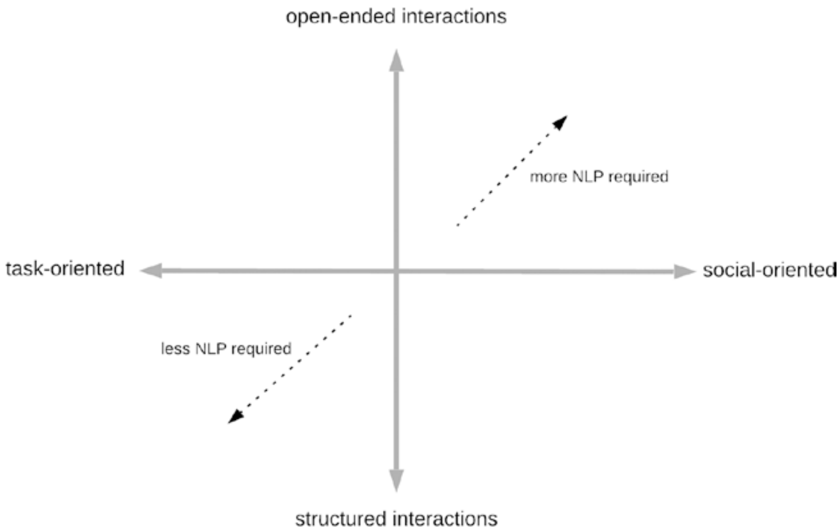

Figure 9-3. Interactions with a conversational application

The more socially oriented and open-ended the interactions are, the more "NLP" (in a very general sense) we will need to keep track of context and maintain a realistic dialog. A good example of such a bot is Microsoft's Xiaoice bot.[4] It operates within the Chinese WeChat platform and has over

[4] https://news.microsoft.com/apac/features/much-more-than-a-chatbot-chinas-xiaoice-mixes-ai-with-emotions-and-wins-over-millions-of-fans/.

660 million "friends." It is specifically developed to be able to act as a companion and display empathy and social skills. The bot has a strong personality and will engage in conversation on any topic. It even has its own TV show!

Alternatively, we may have bots that do allow open-ended interactions, but the object is to support a specific task. Intercom's AnswerBot would fit here. AnswerBot will allow users to type in any open-ended question, which it will try to answer (by diving into FAQs and articles it was provided). If it fails to get an answer, however, or if the user wants to ask a follow-on question on the same topic, it quickly hands over to a human. It will not try to keep a conversation going beyond that point. The aim is to ensure that interactions are either clearly useful or they should swiftly come to an end to allow a human to take over.

What sort of interactions will your conversational application need to support? To what extend do they need to be open-ended vs. structured? What is the bot attempting to achieve in general? In the next section I will present a simple set of considerations that allows you to map the various dimensions of your conversational application, and we will then go on to consider some specific examples of applications.

Planning for Conversational Applications

Considering a conversational application across different dimensions gives us a way of framing and scoping the problem. We can then better identify what methods and tools we will need to solve it. We'll look at five different dimensions and the impact these can have on the application.

Capabilities

The first aspect to consider is what our conversational application needs to do. On one end of the scale we have applications that focus on a single task. For example, a bot that only helps us to approve or reject vacation requests. On the other end we can have applications that behave more like a Siri-like general assistant.

Specialized bots can narrow in on the range of possible interactions and provide more robust behavior. They can keep driving the user toward the appropriate outcome because there are only a few possibilities. More generic assistants, instead, have a challenge just in trying to understand what specific activity the user is trying to complete (as we've probably all experienced with something like Siri or Google Assistant when we wanted to play a song and instead it thought that we wanted to call a contact). The intelligent assistant approach is attractive when you can see more benefits from providing a single entry point, which is what makes them so popular with the large technology companies.

■ If you consider Alexa in this context, you can see its hybrid approach. The keyword Alexa wakes up the general assistant that then, based on what was said, will activate a specific *skill*. Once you are in a skill, however, you are interacting with a single bot that is potentially coming from a provider publishing their *skills* in the Alexa ecosystem—just like an app creator publishes an app on the Google Play store or the iOS store.

Interaction Style

As we discussed earlier, the interaction style of our conversational application can be quite important. Are we looking to support fully open-ended interactions? To what purpose? Is that helping our users to complete a specific task? If a user is trying to describe a complex problem, allowing them to describe that problem in an open-ended way can be amazingly powerful. It is the thing that differentiates conversational applications from glorified interactive voice response systems (IVRs). Everyone hates those because they are simply forcing us to navigate through an endless tree of choices. However, when our application needs specific answers and the user doesn't actually have wide choices, pretending that they can interact in an open-ended way is as annoying. If the only thing a user can say is "yes" or "no," it is pointless to make them think that the possible interactions are more open-ended.

DEALING WITH COMPLEX PROBLEM DESCRIPTIONS

A conversational application we worked on at GreenShoot Labs had the task of helping victims of cybercrime to recover from the attack. The aim of the chatbot was to attempt to understand what type of attack the user experienced and, if successful, provide appropriate remedies. Initial designs of the conversational application centered on complex conversational flows wherein we were asking users a number of different questions to attempt to collect all the pieces of information we needed in order to identify the attack. All those approaches made the interactions too complicated for users. In the end, we decided to ask them to do the simplest thing: "Describe to us what happened."

This way, the users can recount their story just as they would say it to anyone else, and the application then uses natural language processing and a knowledge graph of cyberattack information to identify the attack. It can also ask clarifying questions of the users if the attack type is not immediately clear. This way, the hard work is not passed back to the user by having them answer lots of different questions. Instead, it is the machine that needs to up its game and handle a more complex textual description.

The rest of the interactions with the application, which lead the user to the appropriate guide to deal with the attack and collect feedback, are all structured. By combining open-ended interactions where needed and structuring the rest of the conversation, we are able to keep a focused path through the application and ensure that the user is always clear about where they are in the process.

Context

Context is essential for automation. From a user experience perspective, if people feel that the application does not "get it" they will quickly abandon it. "Getting it" in this case involves making those connections that depend on contextual information and lessen the amount of information we have to explicitly input in an application.

Imagine having to use a car-sharing application in which you can't simply plug in the destination; where you also need to explain where you are currently; and where you have no way of knowing whether the driver is on their way. Payment wouldn't automatically happen. Yes, I realize that is exactly how taxis used to work! All the things that we enjoy about modern car-sharing applications are a result of that application using contextual information to make the task simpler for us.

If users are made to jump through hoops just because applications don't have access to information that is considered readily available (e.g., calendar availability) they will, rightly so, not see the benefit in using a conversational application.

When talking about new products, especially in the context of startups, we often talk about *minimum viable products*. What is the minimum set of features that should be covered for the product to provide value to the user? When you are dealing with automating processes, you can transform that concept into a *minimum viable automated process*. How much of a process are you automating, and is that offering real value to the user?

For example, consider a conversational application that is supposed to help users find a common time to meet. The application does a great job going back and forth between users to find that time, but it does not integrate with the calendar in a way that it can actually create the meeting. The application is released and, to your disappointment, after some initial usage you find that people stopped interacting with it. Investigating further, you realize that the calendar app provides a less powerful but still useful way of comparing everyone's calendar and finding an appropriate meeting time. Since people had to bring up the calendar app anyway to insert the meeting, they just preferred doing everything there.

As such, it is important to consider what the application will need to "know" or "do" is order for it to be able to really solve a problem and offer value. Thinking about the minimum viable automated process and the contextual needs of your application in order to achieve that can save considerable delays and disappointment further down the line.

Platforms

What conversational platforms will the conversational application need to operate on? Will it be just a single platform (e.g., just Slack), or do you require an application that is made available in multiple platforms that might potentially span voice, SMS, Slack, etc.?

You may need multiplatform support to reach more users or because you are bridging a gap between users on different platforms. For example, messages sent via SMS from customers can be piped into Slack for support workers to deal with and then sent back via SMS to the user. Teams that are mobile might be best served by a different application than teams that are in the office.

If you are considering an experience that will need to span or integrate platforms, you will need to consider how well conversations can translate across platforms. If parts of your team are on Slack but a new team that was merged in is still on Skype for Business, what will a single solution for both look like? Each platform enables different types of interactions, and it is up to the designer to find that minimum common denominator that is applicable across both.

Analytics and Improvement

No conversational application will get it right on the first day of launch. While analytics and ongoing improvement are strategically important for any application, they are particularly important when it comes to conversational applications.

There are two dimensions to consider here. First, are the types of "improvements" and "learning" the application might be doing automatically as it interacts with users. Are you confident that the behavior cannot be derailed and that there are checks and balances to allow real-time improvement to happen? Second, if you are planning to do offline review and further training (which is the case for most conversational applications), have you made sure your team will have the time and resources to do this?

I've seen successful deployments of chatbots forced to stop because there wasn't the capability to maintain and monitor the interactions with them. Delegating decision-making to machines can free up resources, but it is best considered as an enhancement to the current team that allows it to scale as they move toward focusing on the harder cases *and* in maintaining the automated software.

Where to Use Conversational Applications

Now, let us start connecting everything that we've discussed so far by reviewing some examples of conversational applications and the implications of introducing such applications into your own conversational platforms. These examples illustrate both what people are already doing today and how that can be enhanced via further automation. These examples are a way to get you thinking about what you could develop and apply in your organization.

Just a note of caution: as we review interesting and fun features, keep in mind that we should not be enamored by the technological capabilities. At every step of the way the hardest question is not if something is possible. The hardest question is if something possible is worth doing. The most important thing to identify is what feature would bring most value to your users.

We will look at examples across three broad areas that conversational platforms are well suited to tackle.

- Notifications
- Monitoring of key data
- Support and collaboration

Intelligent Notifications

Yes, notifications are very likely the least favorite thing for any knowledge worker. There are too many of them, and I have yet to sit in a meeting where someone said they would like to receive more. Nevertheless, notifications are still necessary. Our challenge is, sadly, not to eliminate them but to determine how we can marshal notifications so that we get the most amount of useful information in the least distracting way possible.

The first part of a solution is setting up notifications in a way that we can more effectively control them. This can be achieved by fine-tuning what applications have the right to distract us and by making it easy to tweak settings to fit both our individual and team needs. The second part is to increasingly delegate the decision-making about whether a notification is useful or not (and when it is most useful) to software, so that it helps us be more efficient.

The first part of the challenge is something that conversational collaboration platforms can definitely help with. They are likely the first app we log in to in the morning, as well as the app that we can carry on all different work devices, mobile and not. True to their role as an organizational operating system, they can become the conduit through which all notifications get to us.

In addition, these environments provide us with tools to manage notifications in a variety of ways. One can set hours of the day where they should not be disturbed at all, as well as specific rules on a per-channel basis or based on keywords.

The second part of the challenge is the essence of this book: the use of AI techniques to delegate decision-making to software. In this case the decision-making we want to delegate is when and how we should be notified about something. In addition, it is worth exploring what else might be useful to have a conversational application provide in the context of notifications. One of the limitations of notifications is that it is hard to "action them." We get told something but we are not provided with the tools to do something about it right there and then. With customized notifications, we can integrate the functionality that we need to perform actions as well.

Basic Notifications

The example we will use throughout this section is based on the Slack integration with Google Calendar. At the time of this writing it provided some key notification capabilities out of the box that already proved the value of basic notifications through conversational collaboration platforms.

The first thing a conversational application can do is notify you and allow you to take action directly from within the conversational environment. Through the Slack/Google Calendar integration, if a colleague creates a new meeting you will be notified and can choose to accept or reject the meeting through the notification. As meetings are about to start you can also be notified in Slack, and any relevant information, such as link to join an online video call, is provided in that same context.

Furthermore, since your conversational collaboration platform is where you also provide presence and overall status information for colleagues, the calendar integration edits your status and, as such, indirectly notifies others about whether you are in a meeting or not. This means colleagues can quickly see what you are up to, which can help them decide whether to send you a message or delay it, or whether it's worth getting up and walking over to your desk or not.

Finally, the integration also sends a daily summary at the start of the day with all the events of the day. This means you can stop other applications and e-mail notifications doing the same.

These capabilities are basic and don't involve much beyond usefully taking advantage of the environment and exposing the required functionality. Nevertheless, it is a fantastic starting point from which to go on to build more self-direction into the system.

Active Notifications

Attending a meeting requires a certain amount of preparation before the meeting from a number of different perspectives. How do you get there, who is attending, what is the agenda, what do you need to prepare before the meeting?

The calendar application can act as an active assistant that provides, within the context of the notification, useful information to help you get work done. It can provide travel information, highlight if a room or a location needs to be booked, and provide a checklist of tasks to do in advance.

In order for this functionality to be made possible, our application would need to be able to reason and understand about such things as the location of a meeting, the agenda, checking traffic reports, and more. It should be actively working (through a conversation) with users as they set up meetings, to ensure that it collects information such as the agenda and premeeting tasks. A simple question such as "Would you like to add an agenda for this meeting?" or "Is there anything attendees should prepare in advance?" can help people set up better meetings, help the meeting be more efficient, and establish cultural norms within the team (in this case around better meetings) that improve working life for everyone. A favorite of mine would be something along the lines of "Are you absolutely sure you need this meeting?"!

Team Notifications

Finally, since we are in a team collaboration environment, we can also consider how notifications could be best served if they went out to a team rather than single individuals. What if I could set up a meeting, but rather than mandate which specific people should attend I could request a specific role (e.g., someone from the user experience team, the development team, or finance to help us sort through specific issues)? The notification can then go out to the appropriate team channels and it is left to those teams to identify who is best suited to attend a meeting.

Of course, throughout you need to be thinking how the use of notifications works within the context of a specific organization and its culture and policies. Teams that have a culture of being active and accountable would have a better chance of being able to collectively react, while teams that are more individualistic and focused on their single tasks would not be a good fit for such a feature.

Monitoring Key Data

Tools to collect, collate, analyze, and compare data abound. It has never been easier to create charts and make them available in beautiful dashboards. However, as the chief marketing officer of a large multinational once told me "We've built them, but nobody is coming." Nobody is looking at those beautiful dashboards or, if they are looking at them, there is no confidence in whether the various marketing teams are interpreting data in a consistent way. We have just built ourselves a more elaborate treadmill to run on, but we are still not getting anywhere. What teams need are tools that they can trust to let them know when something interesting has happened and, ideally, explain why that has happened.

Reports typically provide an updated chart or a straightforward piece of information such as "Visits to your site today are 20% more than average." The interpretation of that data is left to the receiver of the notification. Why are visits up on the site? Which section of the site? Is it because a new paid ad campaign started? Is it because a newsletter was sent out? Did our product get a mention in the press? Is 20% above average usual when a marketing campaign kicks in? Should I be expecting more?

There are three obstacles that stop these tools from being more helpful. We need to connect data, contextualize data, and explain data.

Connected Data

First, the tools crunching data and generating reports simply don't have access to all the pieces of information required so as to create a more complete narrative. At best, what happens is that someone will notice a change and a hunt will start across different groups and departments to attempt to understand why the change has happened. This generates questions, interrupts people who were performing other tasks, and can often lead nowhere.

To achieve richer narratives, multiple data sources need to be more tightly integrated and, crucially, the activities *behind* the generated data also need to be surfaced. For example, if you are building a product you should be able to see a timeline of how major releases connect to changes to your key performance indicators.

A path toward more useful activity data collection and integration is through the project management tools or technical code deployment tools. They can already broadcast their activities. If those messages are treated as *activity data sources,* you can start connecting key pieces of information through more careful overall knowledge management. Conversational applications can also assist by proactively reaching out to people and asking for that information, and identifying key moments that could be connected.

Now, don't get me wrong. None of this is easy or simple. It will take effort to marshal the resources needed (not just financial) to put in places applications like this. We will discuss approaches in the upcoming chapters. For now, let's enjoy the possible bright futures.

Contextual Data

We talked about notifications in the previous section and how it is so hard to get them under control. Notifications about key data and metrics is certainly one of the reasons we are inundated with information. There are so many metrics to follow, and we get hit with so many reports, that it all becomes a bit of a blur. The irony is that when we then actually need a piece of data, either because we are discussing next actions with a colleague or are planning activities for the next quarter or we just got asked by our boss about a report, we can't find it.

Conversational applications can help by acting as our assistant to interact with business intelligence tools in order to provide us with the reports we need when we need them—namely, in the context of a conversation.

Requests such as "Can we see sales over the past 6 months for product Y" that lead to a report delivered in a group within out conversational collaboration platform for the entire team to see are within our grasp technologically. However, they require organizations to invest in identifying where they can best benefit from these opportunities. It is no surprise that large tech companies such as Microsoft are directly investing in adding conversational capabilities to their business intelligence tooling. Simply asking for the data we need when we need it in natural language is incredibly powerful and much better suited to how we work and think as humans.

Explainable Data

Finally, even if all the data was accessible and always not more than a few clicks or words away, we as humans are really bad at handling complex, interrelated pieces of data in our head. We are particularly bad at looking at different reports and effectively drawing the links between them.

Conversational applications can play a transformative role, acting as a more helpful interface through which we not only retrieve data but interrogate data. Rather than having to analyze charts to see what is going on, perhaps we should just ask the question:

"What happened today that was out of the norm?"

We could then follow that up with:

"Has this happened before?"

Instead of having to remember how to correlate information and build charts, we can ask:

"Can you compare this data to the same period last year?"

Organizations that are investing now in tooling to provide humane explanations are building the infrastructure for their future and setting themselves up with a significant competitive advantage. The use of natural language generation to provide narratives around data so that users don't need to interpret complex charts and related figures can make access to information far more democratic across an organization.

To achieve these rich narratives, it will be necessary to combine data sources and also capture activities that influence data results in a consistent, coherent, and low-cost way. In Chapter 11, where we discuss AI strategy, we will delve deeper into what it takes to enable solutions such as the preceding one to develop. What could be useful at this point is a thought exercise for yourselves. What would it take to have all the necessary pieces of information in one place within your own organization? If you could construct clear narratives over sets of data (from marketing data to financial data and anything else you need to measure), how much time would that save in terms of having to go dig for the information?

Support and Collaboration

As we already discussed, integrations with conversational collaboration platforms can allow us to complete activities directly through the platform, saving us the need to change context. A simple, yet incredibly effective, demonstration comes with the Google Drive integration with Slack. When a user requests access to a document, the integration sends a notification via a dedicated bot. Within that same notification it allows you to select the level of access you want to give and approve the request. Previously, this action involved finding the notification within a heap of e-mails (provided one reviewed e-mail in a timely fashion) and then accessing the document in order to provide access. In most cases, the person requesting would have to message explicitly, as they were running out of time and needed the information urgently.

Our challenge, from an automation perspective, is to identify how to support more activities and streamline and automate specific junctions in those activities. We will look at three different examples to provide a sense of the level of automation possible.

Automated Helpdesk

With business to consumer interactions, conversational AI is most widely used for helpdesk support automation. There is a clear need, with people wanting support as quickly as possible at any time of day. There is also a clear return on investment, with support center costs reduced and the ability to increase the number of users served.

The same principles can apply to internal support across all levels of the organization: from IT questions—"what browsers do we support?," "what is the Wi-Fi password?"—to HR questions—"what is our policy around sick days?," "who do I contact about issues with colleagues?," and so on. Putting in place a conversational application that acts as the first port of call for internal support means that the various teams can reduce the questions that come directly to them and instead focus on the more complicated cases.

Capturing Decisions and Completing Tasks

We converse so that we can explore a problem, share information, come to a decision, and then plan out how we will get work done. Conversational applications and deep integration between our conversational platforms and project management tools (Trello, Jira, Asana, in-house solutions, etc.) means that we can directly feed that information into the tools that manage tasks and vice-versa.

It can be as simple as being able to right-click on a message to convert that into a task (something that Slack already supports), or it can be a more sophisticated process where the conversational application proactively intervenes to create discussion summaries, identify key decision points, and capture tasks.

Capturing and Sharing Process Knowledge

Being able to capture single tasks and monitor their progress can be very beneficial. What is even more valuable from an organizational perspective is the ability to capture, share, and help enact multistep process knowledge. Any organization necessarily has a number of processes or, simply, "ways it get things done." There is a lot of value in being able to create and curate these processes, even in something as simple as checklists.[5]

Conversational applications can be incredibly effective in helping us both capture processes, curate them over time, and share them more widely. Imagine a scenario where a new colleague is looking to organize a meeting with clients for the first time and is able to call up a checklist with the simple request "How do I organize a meeting with clients?" Then the colleague not only gets the necessary information (here is how you book a room, what to tell reception, how to order food and coffee) but also has an automated *assistant* that will work with them to contact the appropriate people (reception, room booking, security, etc.), send out reminders, and more.

Organizational Collaboration Operations

Conversational applications are the means through which we can connect to the rest of the world and extend our conversational collaboration platform with functionality that is specific to our needs. In this chapter we've discussed the range of issues to consider, as well as given examples of the types of application one could develop.

Something that each organization should now start considering more deeply is what processes they will put in place in order to constantly be thinking about how they can improve and extend their digital environment. Organizations that fail to do so will be less efficient and effective, and far more likely to lose their competitive advantage as a result.

It is quite easy to run a few pilots within any company that demonstrate how value could be created. It is much harder to do this in a consistent manner over time. There are at least two avenues to follow to tackle this challenge.

---

[5] The book Checklist Manifesto by Atul Gawande (Metropolitan Books, 2009) describes how the author **reduced post-surgery deaths by 47%** in hospitals worldwide that trialed the 8-step World Health Organization checklist he created. Atul Gawande describes how he believes there are two types of mistakes: errors of ignorance (we didn't know enough) and errors of ineptitude (we didn't make effective use of what we know). He argues that mistakes in the modern world are most commonly errors of ineptitude. For example, the airline industry has been very effective in minimizing avoidable errors—the knowledge created by previous mistakes has been turned into a practical tool that pilots use constantly. No matter how experienced you are in your field, it's impossible to be infallible, especially under pressure. But you can use checklists to integrate past knowledge at the point where it really makes a difference.

First, it is about creating a culture that supports constant introspection into how things are done and enables people to take issues into their own hands in order to improve the way they work. This is a bottom-up approach that is a crucial precondition to anything else. People have to be open and welcoming to change, and they have to be able to instigate change.

Second, the organization as a whole needs to ensure that it invests the necessary time and financial resources to provide people the space to implement change.

Inspiration can be drawn from the technical space, where investing in development operations (DevOps) is now a well-established practice. DevOps describes the processes that support development teams in creating code and releasing new features. It focuses on providing as smooth an experience as possible, since it recognizes that that will lead to better outcomes. Organizations need collaboration ops that focus on ensuring that people have the best possible experience when collaborating with each other. Investing in conversational collaboration platforms and conversational applications can be a significant aspect of that effort.

PART

III

Building Your AI-Powered Workplace

CHAPTER

10

From Data to Understanding

In 2015, Margaret Sullivan penned a great column for *The New York Times* titled "Awash in Data, Thirsting for Truth."[1] The column asked the question of how important is data to news reporting and to what extend can it obscure or reveal the truth? The answer, as you might expect, was not clear cut.

Lack of data when reporting may lead to people claiming that your story simply does not have the necessary "hard" proof. A focus on narratives of how individuals were affected by events personalizes the story and makes it more compelling and heartwarming. However, such narratives are not necessarily an indication of a larger trend or the experience of the majority.

Overly relying on data may not reveal the entire story either. Instead, it can be accused of ignoring smaller but still very serious issues. In addition, the way you use your data to tell a story may lead you to entirely wrong conclusions.

These are the exact same questions and issues that we have in organizations when trying to define our path from data to understanding.

Despite the possible pitfalls, over the past 15 years we have collectively become obsessed with data. This data-centric thinking is largely a product of

[1] www.nytimes.com/2015/09/06/public-editor/awash-in-data-thirsting-for-truth.html.

R. Ashri, *The AI-Powered Workplace*,
https://doi.org/10.1007/978-1-4842-5476-9_10

Silicon Valley companies, with the likes of Google leading the charge. The classic anecdote that symbolizes their reliance on data is how Google ran an experiment to collect data on what shade of blue (out of 41 candidates[2]) would be most appropriate for a button. A designer who left Google because of these types of behavior said it was an indication of the lack of appreciation of intuition and creativity, necessary preconditions for innovation in design.[3]

Data-driven AI techniques have only exasperated the issue and strengthened the argument of those who call for every decision to be data driven. A favorite statement to throw around in these data-obsessed times is that data is the new oil.[4] It should be treated as a commodity that an organization drills for, extracts, transforms, and uses to generate value for itself. Thankfully, at the same time there are more balanced voices that caution us to not adopt such a data-centric view of the world. In this chapter we are going to follow that more balanced approach to how we can think of data within an organization.

The aim of this chapter is to provide some starting points and useful strategies about how to think of data in your own organization. Despite voicing concerns about overly relying on data, I strongly believe that data is crucial. It is a valuable resource and we need to treat it as such. We should always, however, question and challenge the notion that data is valuable *intrinsically*, outside of a specific context and purpose. This will ensure that what effort we put into data management and curation is directed and efficient.

Giving Data Purpose

In considering the purpose of data it is useful to take a look back at how we data got elevated to be considered the most valuable commodity, and how that position is now being reconsidered or, at least, should be reconsidered.

Being able to store information has always been useful. However, for a long time the potential opportunities were tempered by the cost associated with storing and dealing with data. When data storage was counted in a few megabytes and the devices to handle it took up entire floors in company buildings, you had to carefully consider why you were doing it. As such, retaining data was something most organizations would avoid beyond what was necessary to enable the proper functioning of the organization. The purpose of data had to be very specific, because holding on to it and usefully exploiting it had a very noticeable cost.

---

[2] www.nytimes.com/2009/03/01/business/01marissa.html.
[3] https://stopdesign.com/archive/2009/03/20/goodbye-google.html.
[4] www.economist.com/leaders/2017/05/06/the-worlds-most-valuable-resource-is-no-longer-oil-but-data.

As our capability to store data increased and the cost of doing so decreased, the attitude shifted to one where we would simply avoid the question of what purpose and value specific pieces of data were bringing. We started using terms such as "raw" data. Instead of storing summaries and aggregations we would store every single piece of data. Organizations started shifting into a mindset where they would store as much of it as possible, with the thinking being "you never know how it might be useful in the future, so best to hold on to it." This is also evident in the terminology we use. We purchase "data warehouses" where we keep and organize data, or "data lakes" where we just let data flow in like into a large dam; to be collected and used at some later point. We argue about what is "big data" and what is just "data," with some experts relishing in explaining that what most people think of as a lot of data is simply not that much. We celebrate companies that make no profit because they are accumulating useful data. We talk about how we "strongly believe" that data will prove valuable.

Over the past couple of years, however, cracks are beginning to appear in data's absolute reign. Yes, we can simply store huge amounts of data at marginal extra cost, but we recognize there are other challenges. Storing data does not carry a technical cost but it carries a regulatory and risk cost. With regulations such as the GDPR[5] in place, all organizations are forced to be mindful of why and how they are keeping data. They cannot simply dump it somewhere "just in case" it might at some point be useful. Furthermore, as consumers are getting increasingly more wary of the risks of cybercrime or other forms of digital manipulation (e.g., shaping opinions for political gain), they are looking for organizations that can prove they are worthy custodians of their data. Data carries with it opportunity but it can also be a liability, so clearly defining the purpose of data is once more important.

Now, if holding on to data needs to serve a purpose, what is that purpose in the context of AI-powered applications? Well, to start with we need to define what sort of AI-powered applications we are looking to build. What decision-making do we want to delegate to machines and what model of how the world works will we need to have in place in order to enable that decision-making? Based on the model we need and the way in which we can go about creating it, we can figure out what types of data we should be using.

As we discussed in Chapter 3 there are two ways for us to uncover the necessary decision-making model that we would need to implement in order to automate processes. We can use our own knowledge of a domain to develop hypotheses and create appropriate models of behavior, or we can look for patterns in data and use algorithms that will explore that data in order to discover a possible model.

[5] GDPR is the EU General Data Protection Regulation, which represents a step change in data regulation in Europe. It describes how organizations are to treat user data and penalties they may face in the case of a data breach or a failing on their side to properly handle user data.

For the latter, data serves a very clear purpose. Data is the terrain that we explore to uncover correlations. If we want to be able to build a document classifier of different types of documents, we need lots of examples of those different types of documents. If we want to uncover patterns of behavior for our teams, clients, or partners, we need data that captures that behavior. But does this mean that we should store every little piece of data potentially available to us?

That can be quite a hard question to answer outside of a specific context. My guideline right now would be to err on the side of caution and only increase your data retention capabilities alongside an increased maturity of your data governance capabilities. In other words, if you are just starting out and exploring how you could use AI techniques, don't start by mandating that every single piece of data should be stored for potential future exploitation, no matter how tempting that may be. Start by building out structures within the organization that can ensure that you have proper data governance in place. Start by building your confidence that you are doing the right things when it comes to employees and clients and how their data is treated. The more confident you are about how well data is managed, the bolder you can be about how much data is stored.

For model-driven techniques it may feel that data is less of an important element. We are, after all, not going to require huge amounts of data in order to build our models. Don't forget, however, that those model-driven techniques are there to manipulate data. They may require less data, but the data they will need requires more structure. For example, let's suppose you've built an ontology that describes how the different types of products you produce relate to each other and the skills that are required to develop them or the materials that are required to produce them. In order to use this ontology to power queries, you would need a database where these different pieces of information are appropriately stored and annotated. To achieve this you will need a more disciplined approach to data management that ensures, for example, that as data is updated it continues to conform to your structure's data needs.

Whatever AI technique you use, the ability to *govern* data is crucial. To get that in place you need a data strategy. In the next section we look at some practical methods to employ.

Developing a Data Strategy—from Theory to Practice

Something that I often come across as I sit down with teams to advise them on their data strategy is that initial conversations or workshops are split into two phases.

There is the first phase, where the "important" stakeholders like to get involved and dedicate some of their precious time. Here we get to define vision, goals, and "high-level" strategy. Things are upbeat in this phase. The world is going to be amazing and everyone is excited.

Then in the practical second phase, once "they" leave the room, the practicalities of getting things done start being discussed. Here things quickly go downhill. In organizations that are not tech-driven and transparent in their methods from the start, simple tasks can become incredibly challenging. This is not because of lack of will but because of how complex it is to navigate the various layers of responsibility and departmentalization of data in order to get to something useful.

I will work through an example as a way to highlight some of these issues and put in place strategies that are cognizant of these.

The Challenge

The company is a large engineering company that builds complex widgets for clients. It spans several geographical sites and services multiple industries. It has highly specialized personnel, and projects are considered mission-critical for clients. We are looking to build a tool that will support project managers in doing the initial planning around putting together teams, schedules, and raising any other considerations that may be relevant as the company is gearing up to take on a new project.

Projects always require a variety of different skills, are expected to span several months, and will occupy different roles in different ways throughout their lifetime.

Given just this brief outline, if you had to do some initial planning for a challenge like this, what types of knowledge would you need? Here is my tentative list:

- A list of who is in the organization and what their skills and past experiences are—this will allow us to figure out who might be a good fit for a given project.
- A view into their day-to-day calendars in order to set up some initial exploratory meetings
- A view into their longer term commitments to understand how viable their participation in the project is. There are bound to be several combinations of suitable teams, so we need to have the teams discuss and square that against their long-term commitments.

- The ability to search through past projects on the same domain for any potential areas of concerns—perhaps something someone documented in the past that could inform our current actions

This is, of course, a very high-level list but it gives us enough to explore what strategies we can put in place to tackle the task from a data perspective. Let us take each requirement in turn and explore what demands it places on our data capabilities.

A Data Strategy in Support of Communication and Collaboration

In order to know who is in the organization and what their skills are, we need an up-to-date database of people in the organization. That is obvious enough and one would think it should never be a big ask for any organization. A list should exist somewhere! Everyone presumably gets paid at the end of the month so, worst-case scenario, Finance/HR will have something.

Well, the list does exist but, as it turns out, it is not accessible in a way that you can build an application that uses it. The only way you would get it would be in printed format. It turns out that the HR system is quite outdated and only works on their machines. The IT department absolutely refuses to provide any access, as that would require skills that they simply don't have anymore. They cannot guarantee the safety of data if they start tinkering with the system, so they simply leave it as it is. It's not all bad news though. We learn that HR is planning to replace the current system with a new one—the new system is going to be rolled out in about 6 months.

Being the optimist that you are, you see that as an opportunity. "Fantastic," you think, "let us make sure that the new system is going to be able to provide us with rich profiles of the people within the company so that we can build AI applications that can answer complex questions about who is able to do what." Nice idea, but it turns out that HR has finished the requirements spec with the vendor, and they can't change what was already planned now. Contracts have been signed and development has started.

You are not one who gives up easily though. You push back, get people in meetings, and passionately argue your point. You explain how this system is going to hold data that is valuable to the entire company, not just for HR purposes. You argue for strategic thinking that ensures that the company is thinking about data in a more general sense and not just within the context of specific projects. You explain how every system that is the source of data needs to enable other systems to easily access that data.

Finally, it clicks for everyone. A more general vision and strategy for data is required and should be developed in support of the organizational vision and strategy. Organizational goals such as "renew HR capabilities" and "improve project planning" are not isolated. Their data needs intersect.

■ A principle for data management should be that data is never locked into a single system; it has to be accessible by other systems.

Having multiple parties being able to access the same data is a great step in the right direction. Ensuring that that data is always accessible to your organization even as you may change vendors of software is a sensible insurance policy always, irrespective of your AI ambitions.

Coupled to ensuring that access to data is possible is ensuring that the way data is described is widely usable. Teams should work together to develop common models of how to describe key aspects of organizational behavior such as roles and skills of people. That starts giving you superpowers. It can be a point of differentiation from the competition because you have a better handle on all the different things your teams can achieve. The better your knowledge representation capabilities, the more sophisticated your AI-powered planning tool can be. To achieve that you need teams that are all switched on to the power of data and talk across team and departmental boundaries to coordinate and uncover opportunities. It is not a one-off thing either. It requires a culture change that ensures that these conversations keep happening. Just as everything else is subject to change, so are the ways you will use to describe things.

■ Make sure that discussions around the handling and description of data happen consistently across the entire organization, to enable you to capture opportunities and identify where multiple stakeholders should have a say into how data is described.

A Data Strategy That Enables Connection and Aggregation

The next challenge is about accessing people's calendars and understanding their day-to-day commitments. Your optimistic side kept telling you that this shouldn't be a huge problem. Calendars are "standardized" technology, so surely you would be able to simply get access to it all.

However, it seems that calendars aren't quite as standard as one would think. The company recently acquired two smaller companies and their infrastructure is very different. They don't allow any "external" systems to access calendar information. You discuss the possibility of moving them to the common platform, but that triggers a cascade of dependencies on other systems they should update. There is no easy short-term solution. It will take time for everyone to align on the same systems. At this point you are faced with two options: give up on being able to integrate calendar data or tackle the problem in a different way.

Giving up, as you've already proved, is not what you do. You argue for building dedicated tools whose only task will be to act as a bridge between the two systems. They will extract information from the internal calendar systems and provide a safe way for external systems to access that information and make it more widely available.

This was another aha moment for the wider organization and something that can form another pillar of the overall data strategy. Sometimes change cannot be forced through alignment on the tooling used. It isn't because of unwillingness to do so. Change sometimes need time and there are only so many things that you can change at once. A new acquisition has a number of issues to work through. Not having to worry about upgrading their calendar systems immediately might just be the point to let go.

A better way to deal with the issue is to adapt to the situation with tooling that allows you to transform and aggregate data. It will be more costly in the short term, and it will mean you are building tooling that you will eventually throw away. But you have to balance that against the organizational cost of forcing people to move too quickly on too many fronts, and the opportunity cost of not having joined up data as early as possible.

■ Better handling of data also means supporting the creation of tooling with the explicit purpose of extracting and transforming data. At times it is simpler to remove the data management problem from the department, team, or specific software that holds the data and instead build a tool on top of that.

A Data Strategy to Unify Information through Standards

The next set of requirements is around understanding people's long-term commitments, both historically and in the future. Once more, the optimist in you was hoping to find a single centralized tool everyone uses to plan their schedule that would give you all the information in an easy to understand way. Reality, as ever, is very messy. There is no single way that people manage their availability and commitments. Project teams tend to pick a tool for a specific project, and there is a lot of heterogeneity across teams and across projects.

The challenge this creates with data management is that data is lost as it is spread across a variety of different tools, and there is no overall unifying approach to it. At the same time, it is important that teams are empowered to choose the best tools they want for specific projects rather than being forced to a single tool that may not always be the best fit. Do you force a single tool and eventually get the data you need but risk upsetting the teams, or do you resign to the fact that you will not have access to this data?

This is where standards can play a role. You don't have to impose a top-down solution; you can simply mandate that certain capabilities should be met by any tool that project teams pick. For example, in the case of planning and resource allocation data, a standard could stipulate that any tool used to capture where time is spent is able to export its data in a format that provides the information the organization needs to plan. This means you would have to define a format (even something as simple as an Excel spreadsheet with specific fields can do the job) and provide teams with guidelines and ways to check that their chosen tool would conform with and output to that standard.

Introducing standards across a variety of activities allows teams to still move with a certain degree of autonomy, creates a useful discussion around what should and shouldn't be included in the standard, and has the effect of getting everyone considering the impact of storing data in specific types of tools.

There is value is being able to change tooling and meet the specific needs of a project or team, but there is also value in consistency. Standards provide a way to navigate those two aspects.

A Data strategy to Iteratively Improve Tooling

The final requirement is around being able to develop the company's capability to extract learnings from past projects in order to be able to inform future projects.

There is a trove of information about past projects, but it is all spread across different systems and there is no way to search across everything. In addition, there are different file formats and different conventions used within the files. We would need to collect it all together, place it in a search engine of some form, and build an interface that would allow us to search through it effectively. Furthermore, in order for this search to be efficient, we need to be able to search the documents using generic terms that are appropriately expanded to terms that may be related to what we need but that we are not necessarily aware of.

For example, if someone searches for projects that involved "metal-joining" the search engine should be able to expand that to "soldering," "brazing," and "flux," since those are related terms. To achieve this, you would have to settle on a company-wide way of describing all these different things and then build tools that would classify documents appropriately so as to power search.

At this point everyone is starting to realize that this is not going to be a simple task, and they are baulking at the potential scope and cost of it. Once more you need to gather the various stakeholders and produce a convincing plan that will keep everyone on board. The key here is to break the problem down into manageable phases, with each step delivering value so that it can justify the next step of investment. In the best tradition of start-up culture, you need to think big, start small, and scale quickly.

You explain how you can start quickly by collecting data behind a standard search engine. It may not provide the full benefit of an NLP-powered engine, but it is a more manageable task that will deliver immediate value. At the same time, a cross-functional team is created to start developing appropriate models of the type of knowledge that a more powerful search engine can start manipulating. Finally, company guidelines can be developed to ensure that every project adds to their list of debriefing tasks the transfer of project knowledge into datastores that can be accessed by the search engine.

At a second phase you can start applying automation via the introduction of NLP to better understand the types of documents available and make search better for users. You will integrate this functionality with your collaboration environment so that people can run searches through natural language questions. It is now starting to look like a fun project to do and one that is actually manageable. While challenges will always exist, there is less initial fear to get things started.

Data needs to be prepared in order for it to be useful, and projects need to be broken down into iterative steps that provide value at each phase. As with any new venture, take a think big, start small, and scale fast approach to problem solving.

From Data to Understanding

Ultimately, everything that we do around data is there to lead to understanding. Nobody should be proud of how many terabytes of information they have stored and how much computing power they are expending to analyze it.

There is no intrinsic value in that. There is only value in your capability to use data in a way that will provide you with insight and enable you to get things done.

In this chapter we looked at both the high-level purpose of data for data-driven and model-driven techniques, and discussed how increased use of data to power automated decision-making has to be accompanied by increased confidence in data governance.

Finally, we looked at a few different strategies you can employ in order to solve real-world problems. Deciding to tackle a single problem and using that to inform your wider organizational strategy is a great way to get started. Learnings will always vary from team to team, and it is important to gain practical experience. These bottom-up learnings tied to top-down willingness to invest can provide a winning approach.

Once a first problem is successfully tackled, the whole organization gains more confidence and the process can be repeated until it becomes part of standard procedure rather than a one-off experiment. It is only then that real transformation happens, when the process is embedded in the culture and is no longer a novelty activity.

CHAPTER

11

Defining an AI Strategy

In 2017 at the annual Google I/O conference, Sundar Pichai, Google's CEO, stood up on stage and announced that the opportunities afforded by AI were such that Google was moving from a mobile-first to an AI-first strategy. That statement really brought the message home to many about the importance of AI and the need to formulate an AI strategy. Google, the technology behemoth, was turning into a full-blown AI company. They had alluded to it in the past; everyone knew that Google was heavily invested in AI and using it extensively, but this was different. It left no space for doubt. AI-first.

If we stop for a second and analyze it, what does the move from mobile-first to AI-first really mean though? Well, the mobile-first strategy was one that recognized that the primary way people access digital services is through mobile devices. As such, any product Google produced needed to work on mobiles, first and foremost. A desktop-only product was simply not an option. Google's hypothesis behind the mobile-first strategy was that if it provided the smoothest and fastest mobile experience, users would flock to their products. The challenge was that mobile-first development at the time required a focused effort and presented technical challenges. It needed support from the top, as it created additional product development resource demands. That Google strategy soon became the norm across technology companies. These days it is not a *strategy* to say that you are mobile-first.

R. Ashri, *The AI-Powered Workplace*,
https://doi.org/10.1007/978-1-4842-5476-9_11

Mobile-first is simply one of the sensible pillars of anything digital and not a huge achievement. There is no need to have it be the defining, outward facing strategic statement of the company. It's business as usual.

With the 2017 AI-first announcement, Google was recognizing publicly that there was a new opportunity on the horizon and new challenges to overcome. The opportunity revolves, once more, around meeting users' expectations and demands. Users are now savvy digital citizens. We all expect a whole new level of sophistication from our devices, especially as we catch glimpses of how things can work better and are no longer mystified by digital services. It's not enough to simply have access to the service from any device or for that service to be reliable. The more technologically savvy tech companies are demonstrating how smart interconnected services can be and that, in turn, leads us to want all our services to work in the same way. When you can order any meal and have it show up at your front door in 20 minutes or buy any item and have it delivered that same day, it creates a certain set of expectations. This in turn creates a feedback loop that forces companies to upgrade everything else that they do. To stay ahead of the competition and satisfy users, services have to be far more aware of our context and our needs and react appropriately to them, or even proactively anticipate them. If an app doesn't "get it" we are quick to call it useless, complain in frustration, and move on to the next one. The path to more user-friendly services though is paved with AI techniques. This is what an AI-first strategy means: recognizing that the next evolution of your products and services will depend heavily on your ability to enhance them through the use of AI techniques. It also acknowledges that integrating AI techniques will require concentrated effort and focus, to overcome some of the challenges that will inevitably present themselves. It is not business as usual.

An AI-first strategy acknowledges that the path to smarter and better services depends on an organization's ability to effectively exploit AI techniques and capabilities.

It's not just Google, of course. Every major technology company has a significant AI effort underway, transforming both how it does work internally and the products it offers. IBM has been promoting the IBM Watson brand since 2010, even briefly turning it into a household name in 2011 when it won the *Jeopardy!* TV game show in the United States. SAP has the Leonardo platform, what they call an all-encompassing digital innovation system with machine learning and other AI techniques at its core. Salesforce has Einstein, an AI platform that weaves AI across its CRM offerings. Amazon is arguably as advanced as Google in considering how AI can transform every single aspect of their business. With Apple the strategy is all-encompassing, with services being enhanced through AI *and* their hardware evolving to better support AI through dedicated chips able to speed up AI-specific computations on devices.

One can't help but feel somewhat overwhelmed by the efforts of the technology behemoths. The effort has reached such a fever pitch that universities are complaining that they can't staff their own AI research groups because companies are hiring people as fast as they can find them.[1]

How can other organizations develop their own thinking around AI in straightforward terms? Are we all supposed to move to an AI-first strategy and join the race? Do we need to start hiring AI-experts?

One of the main aims of this chapter is to give you a *practical* approach to developing your own AI strategy. It's not about joining the AI race (unless there are extremely good reasons to do so). It's about identifying the principles to adhere to and tactics to apply at different stages and for different needs for your particular journey, so that you make sure you are making the most out of what AI technologies can offer your business.

A Practical Approach to AI Strategy

> *"[Strategy is] strength applied to the most promising opportunity."*
>
> —Richard P. Rumelt, *Good Strategy Bad Strategy*[2]

Developing a strategy for anything is an incredibly challenging task. The only information you have to act on is what has happened in the past. Everything that will happen in the future is, ultimately, a guess. In the same way AI techniques work, a strategy depends on you having a model of how the world works, based on the information you've had so far, and devising a set of rules that will allow you to make a prediction about the future. However, unlike the problems AI can currently help us solve, you need to deal with an almost unlimited set of variables. To make matters worse, there is no one-size-fits-all solution. You have to craft the strategy that is right for your organization, and the best strategy for your team will, almost by definition, be an unsuitable strategy for a different place.

As such, there is no single true recipe that will lead to your ideal AI strategy. Only you know what the real ingredients are. What I can do is talk about general principles, ways to prepare and specific methods to apply. Like a chef in a kitchen presented with a mystery box of ingredients, it is your task to open the box, figure out what you have available, and make the best possible use of each ingredient.

[1] An iconic example of this was when Uber gutted the CMU Robotics lab by hiring 40 of the lab's top researchers (out of 100), including the director.
[2] Crown Business, 2011.

Principles

Principles for a strategy act as guiding stars to help us choose one direction rather than another when the ground truths don't clearly point to a choice. The principles I present here are the distillation of numerous conversations with large and small organizations about what has and hasn't worked for them, and what is important to focus on and what instead is more of a distraction.

Think Big, Start Realistically, Scale Appropriately

Those of you familiar with the lean startup movement will see the seeds of that movement's mantra in the heading. It usually goes: "think big, start small, scale fast." The lean startup movement was started by Eric Reis, who, through his book *The Lean Startup*[3] described a methodology for doing business that embraces uncertainty and minimizes risk by testing the viability of business models through the execution of carefully managed and cost-effective experiments. The experiments are designed to test the most critical hypothesis behind a new feature or product before scaling that product to full-blown production. For lean startupers, it is important to have a big far-reaching vision but then start with small experiments that are not that costly, before moving on to scaling as fast as possible in order to capture market value. A strategy for applying AI techniques to your workplace can benefit from the same sort of thinking. At the same time, there are a couple of things to keep in mind—which is why I distorted the mantra to "Think Big, Start Realistically, Scale Appropriately."

Think Big

The potential of the application of AI techniques to how we do work is transformational. It is necessary to think big in order to understand the true scope and assign it appropriate importance. Just like Google with their public AI-first strategy statement, it is about giving the entire mission enough importance so that people pay attention and prioritize it appropriately.

Thinking big is also about recognizing that automation is not just about more of the same but at a lower cost or higher speed. Don't think of the introduction of automation in a business as akin to a factory line, where component A can be attached to component B in a faster and less error-prone way. Automation through AI also enables entire new ways of doing things. It enables new business models. It is a circle whereby the need for automation leads to digitalization of information and process, which in turn generates

[3] Eric Reis, *The Lean Startup* (New York: Crown Publishing, 2011).

more information that can be further exploited and leads to completely new ways of solving a problem.

For example, automating support handling doesn't just mean that you will need fewer resources to deal with the same amount of support requests. It also means that you can better understand support issues as each support request is carefully analyzed by increasingly more sophisticated natural language-understanding capabilities. The resulting data can be fed directly into product development, and the results of product improvements can be traced back to the types of questions that come through support after a release. It means that you can evolve your support teams to become customer success teams that engage with clients in a completely different way. The introduction of automation in support handling can affect the business all the way through to how product design and development take place and customer relationships are managed. However, to realize the full benefits of the effort, you need the big vision version so that you put in place the right stepping stones at each phase to be able to get to the top of the hill.

Figure 11-1 illustrates this flow for any process. We start from a place with historical data and understanding of our processes (1) and do the work to augment our standard processes through automation (2). This, if done carefully and with a mind to the future, can lead to enhanced data and understanding (3). This enhanced understanding starts creating a virtuous feedback loop that can go back into augmenting existing processes (2) but also into creating additional opportunities (4) and even new processes (5).

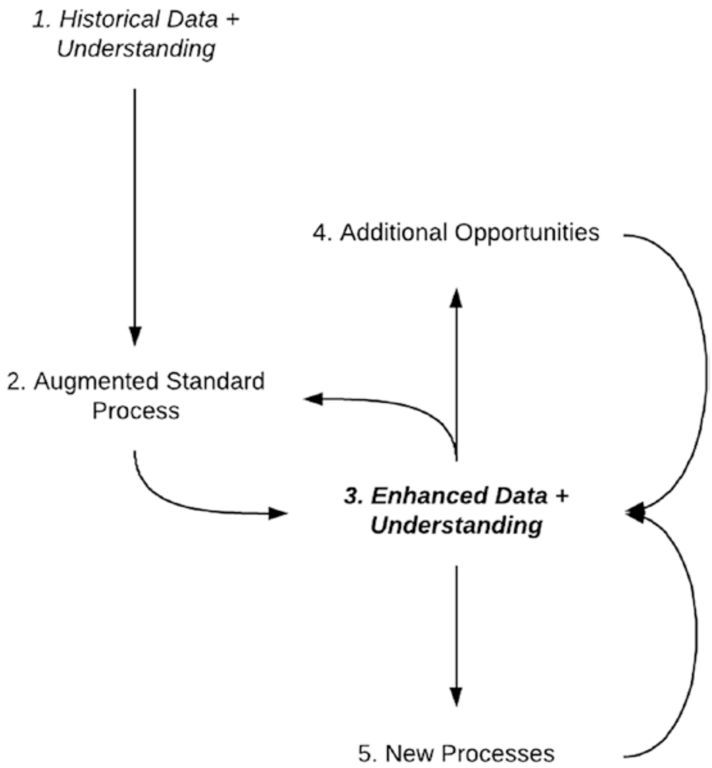

Figure 11-1. The AI virtuous process enhancment flow

Start (and Evolve) Realistically

Applying new technologies always involves a high amount of risk. It is not business as usual. You need to be able to absorb and react to a lot of learnings, and things will not turn out as you would expect them. The lean startup approach has made it popular for people to push to start with small experiments (minimum viable products) that will test a key hypothesis before committing additional resources. Overall, that's a very sensible approach. The crucial question, of course, is exactly how minimal a minimum viable product can be. It has to be enough to give you information about whether it is worth scaling the entire effort. If the product is too simple and too minimal, it is not going to be useful.

With AI techniques we are in a very similar situation. If you are attempting to deploy a prediction algorithm, you need to ensure that you are giving enough space for people to prepare enough data and try enough approaches to have a firm understanding of whether the problem can be solved. Think of it as the equivalent of the escape velocity for a space launch. Engineers know that unless a launch missile is able to develop a velocity of 25,000 km/h it will not be able to escape the earth's gravitational pull. If you know that your investment in engineering will not be enough to surpass that minimum threshold, you are better off spending that money elsewhere altogether.

Now, recognizing where that threshold lies requires a combination of experience and willingness to have some false starts. As such, I believe that there are two key tests that a plan must pass. The first is to ensure that the individual steps in the roadmap have enough clarity that there can be some preliminary research into their feasibility, and that no step is so big that if it fails there is no space for a change of direction. The second is to ensure that each step is fed the appropriate level of inputs, a sort of readiness gate that ensures you are doing enough for it to be useful without doing too much. If coming out of one step you do not meet the readiness gate to move into the next step, you can stop and reconsider.

Scale Appropriately

We, of course, want to scale fast. A large part of the appeal of automated decision-making is that it can enable organizations to move much faster.

However, it is precisely because of the ability of automation to scale quickly and the nature of the automation taking place that we need to be cautious about the speed at which we scale. Automation, in this case, depends on use of past data and knowledge to encode the rules of how a certain aspect of the workplace operates. The resulting software program is only as good as our own understanding of our problem and the data that we used. Before scaling from one thousand to ten thousand users, or from one geographical location to another, we need to ensure that the underlying assumptions remain the same. Rushing to scale an automation model beyond the confines of its test environment without checks in place to ensure that it is behaving as it's supposed to can lead to unintended consequences, such as exhibiting bias in its decision-making or introducing errors that are hard to catch.

Tangible Outcomes Matter

An AI strategy, like any other plan, needs to take into account the wider environment within which it will develop. Too often anything new is cornered off and placed with an innovation lab where it will be studied and experimented on, but where it faces a hard time to see the light of day in the real world of the business. This is especially true of technologies like AI.

Innovation labs are not, by default, a bad idea. Organizations that can afford to set up dedicated teams to experiment are lucky, and they should fully take advantage of the possibilities. The learnings of innovation, however, need to be put up against the cold, hard light of the real situation. You are not solving the entire problem unless you have dealt with the entire reality of it. It's like saying you want to build a rover to explore Mars, but you only test it on your local neighborhood roads.

An AI strategy that does not plan for how a technology is going to move from the cocoon of an innovation team to the actual business is not a complete strategy. This involves convincing stressed business units with day-to-day operational priorities that they should adopt new technologies. It involves ensuring that the solution brings measurable benefits that will make a difference to a division's budget. It involves dealing with concerns of staff that they are going to be replaced by automation, and putting in place training programs to deal with the change in the way things are done. If those elements are not there, you have no real measure of the success of the effort.

An AI strategy Is Not About AI

This heading is counterintuitive on purpose.

An AI strategy is not about using AI at all costs. I've seen enough companies get lost in trying to make AI work no matter what, that I feel this really needs to be driven home from the very start.

As we said repeatedly in this book, AI is useful in helping us delegate decision-making to machines. Also, as we said in the intro to this chapter, AI helps us meet user needs and expectations. AI, however, is not a goal in and of itself. The objective is never and should never be simply to "use AI." The goal of an AI strategy is to create the necessary preconditions and processes that will allow an organization to

1. Determine whether AI techniques are applicable and can help solve problems in a better way.
2. Ensure that the organization is in a position to exploit the opportunity, provided that AI techniques are applicable.

It is perfectly OK if the outcome of the process is that the use of AI techniques is simply not a good idea for solving a specific problem.

This is exactly what happened in a particular conversational AI project. The organization was looking to implement a chatbot to handle queries from clients who had issues with their travel documents while not in their own country. Effort went into thinking about what the appropriate language understanding solution would be to identify what happened to the travel documents (lost, damaged, stolen, etc.). However, when it came to understanding what the appropriate response should be for each type of problem, they realized that it was always the same. No matter what caused the issue, the way to deal with it always consisted of filling out the same form or getting in touch via phone for urgent cases. All the up-front effort in recognizing the cause of the problem was not required. This was a problem that could be resolved with improved information architecture on the web site, and did not require the use of any AI technology.

There Is No "True" AI Test to Pass

Similar to the previous principle, this one is a result of another common pitfall. When people embark on AI projects, one of the main concerns is to do something that is "real" AI. In many ways, it's only natural to have this concern. We are dealing with new technologies with definitions that are incredibly fluid. People feel they need to ensure they are doing the "right thing" in the face of a lot of conflicting information. Whether hiring or looking outside for help, organizations want to ensure that they are not "cheated" with "fake" AI.

The result often is that solutions to problems are overengineered, or perfectly suitable solutions are discarded because they don't meet some unclear "AI" test. When dealing with virtual assistants, this often involves discussions around what is a sophisticated enough conversation that *feels* human-like. Simple, to-the-point conversations are replaced with open-ended conversations that would mimic natural language more in order to meet this human-like standard. With prediction algorithms and machine learning it is often about discarding advice that calls for the use of standard and well-established techniques from statistics in favor of techniques that involve the direct use of deep learning because deep learning *feels* like truer AI.

The challenge with these issues is that they are hard to deal with. The end solution works. A problem has been solved. But it is a much more brittle solution because more AI technology was imposed than what was really required. Everyone feels excited because it feels more futuristic but, in reality, they are only setting themselves up for more pain as the solution evolves.

A comprehensive AI strategy needs to include checkpoints where an honest discussion is had about whether the solution is the best one for the problem at hand, or whether it is simply a solution that satisfies the need to demonstrate the use of AI as opposed to reaping the true benefits of the technology.

Decide If You Really Want It

Talking about how things ought to be is easy. Nobody can disagree with those slick diagrams that have the user at the center with concentric circles spanning out, neatly featuring words such as collaboration, communication, and connection. Bringing about actual change is hard though. Reality very rarely matches these idealized approaches. Reality is messy, with layers of processes, data, tools, and people having evolved and changed over time. Reality also doesn't take a break. You can't hit the pause button, figure out what you want to do, put it in place, and then hit play again. Decisions will be interconnected, and the current state and future state will need constant understanding and untangling.

It's for this reason that the biggest challenge in realizing an AI strategy (and any other strategy really) is to get firm commitment from all stakeholders that it is a mission worth going on. That includes you, the AI pioneer in your organization. Both at a personal and organizational level, there needs to be acceptance that it will be a long and complex process to get to a place where the digital workplace begins to align with what your vision and mission are. There are no simple solutions, no silver bullets, and it's not about how much money you can throw at the problem (although obviously resources will be needed).

As such, the first real question to ask is whether you, individually, and the entire team, as a group, want to embark on the journey of changing and improving things. Depending on the organization it may start out as a lonely trip, and you may need to state your case multiple times to different stakeholders that all need to sign up for the effort. However, as cliched as it sounds, deciding that the journey is one worth taking is the most significant step.

Methods

The previous section gave us some principles to help us judge and steer our plan. The next step is to look at some more specific methods that will help us to get started and realize a plan.

Find Your Place

In developing a plan that introduces automation through AI techniques you will always end up asking three very specific questions.

1. How do we currently solve a problem? What steps do we go through to deal with an issue, and where are the rules that define the process?
2. What data do we have about the problem? What historical data do we have and in what condition is it to help us better understand the problem?
3. What current activities are taking place that would enhance or hinder our ability to use AI techniques to solve a specific problem?

How Do We Currently Solve a Problem?

Mapping out processes is a fundamental task of any organization and there are numerous techniques that can help with that, from business process modeling to data flow modeling. It is not the task of this chapter to provide an introduction to those techniques. Instead, it highlights some of the issues that are relevant to understanding processes with a view of applying AI techniques to those processes.

Uncover the Real Process

One of the first lessons that most customer service support automation problem solvers learn is that the way people on the front lines deal with issues will differ from what the manual describes, and it does so for a very good reason. The manual is wrong.

The humans, as the intelligent and extremely adaptive beings that they are, have figured out all the shortcuts, hacks, and workarounds to the processes to make them actually work. They've crossed out the wrong information in the manual, stuck a post-it note on their screen with the fix and moved on. When we attempt to introduce automation, we need to ensure that *that* knowledge is captured and included in the automated process.

Automation means spending time with the people actually doing the tasks, to learn what the real process looks like right now. This will give invaluable information about what can effectively be fully automated and what can be done to augment and assist the people involved in the process.

Simulate Processes with Humans

Whether attempting to automate an existing process or looking to introduce a new one, it is useful to consider whether you can simulate the process: put simply, whether you can fake it using humans. As tedious as it may seem, I would advise you to get a volunteer who will *be* that process to start with. Have a human sit behind a keyboard and *pretend* to be an automated procurement information service answering questions about the status of various invoices from across the organization.

This gives you invaluable information about how people will interact with an automated service without having to build the service itself. Those interactions can in turn inform how the service gets developed and uncover issues around the integration of the service into the organization. Dedicating a few days to learn how people are likely to use the service is the most cost-effective way of doing it.

Don't Ignore Experience

Of course, not every solution can be simulated by a human being. If you want to identify the best sales prospects to contact based on data analysis of information in your CRM, you will have to do that data analysis. You should, however, introduce the subject matter experts as soon as possible, to interpret the data and see if their intuition matches what the data is saying or whether there is a significant mismatch. If there is a mismatch it needs to be examined. It doesn't mean the data is wrong, or the experts are wrong. You simply need to recognize that data analysis will uncover correlations, but not all correlations translate to actual valid causal effects. An expert will be able to smoke out some of the more obvious misleading results.

What Data Do We Have?

We've already discussed data in the previous chapter. One issue we left out, though, is mapping out what data is currently there. To provide a simple framework for describing your data, I will borrow from ideas that were developed by the open data movement. Tim Berners-Lee (yes, that Tim—the inventor of the Web) suggested a five-star scheme for describing open data[4] to share publicly. However, it is also a useful guide for describing data within an organization, especially as we think of data being shared between different groups and departments.

One-Star—Data Available for Use in Whatever Format

The first step is to simply have data available for use in whatever format is possible, in a way that is accessible to the wider organization. The test for one-star data within an organization is that people can actually find it and are able to trace who is responsible for it and what rules govern access to it. The data is very likely to be unstructured data in PDF documents, but at least it is findable. You will need to deploy more heavyweight techniques such as text mining to extract data.

Two-Star—Data Available in a Structured Format

A step up is to have this findable and attributable data in a structured format that is more machine friendly. It could be an Excel spreadsheet, for example, as opposed to a scan of a table. Structured data means that it is easier to get to, but you may still be dealing with file formats that are not in use by any software right now or where the schema behind them is not well understood or documented.

[4] https://5stardata.info/en/.

Three-Star—Data Available in a Well-Understood Structured Format

Three-star data is data that you can access in a well-understood structured format. We may be dealing with a CSV (comma-separated values) file or a database table. You will still need to uncover information around the schema and access to the data.

Four-Star—Data Available via Documented API

In this case we are dealing with structured data available via a documented and well-maintained API (application programming interface). This indicates that there is a team on the other end that is curating access to the data and that more well-thought out data governance processes are in place.

Five-Star—Data Linked Across Sets to Provide Context

With five-star data we are not only able to gain access via a well-documented API, that data is interlinked to other data sets within the organization, allowing us to make more interesting inferences about the context of the data.

Discovering and ranking datasets provides a map that can indicate where the best starting point is. We can start from where data is of the best quality to prove the value of AI-powered automation and motivate the improvement of the rest of our datasets.

Connect Activities

The development and execution of an AI strategy should not be viewed as something that happens in isolation to other activities. It is crucial that it informs thought from the very start. You could view AI capabilities as simply a toolbox you reach into and pull out useful tools to help solve problems as they appear. You could argue that since AI is a technology, a way of solving a problem, it doesn't need to feature when defining overarching strategies. Only once you get to the point where you need to build a solution do you start exploring the space of AI techniques and capabilities to see what applies to your problem.

I think that approach is flawed. It fails to capitalize on one of the most significant aspects of strategy: the orchestration of activities so that the whole is greater than the sum of its parts. An AI strategy cannot and should not stand in isolation from your wider digital strategy, which should be connected to your overall strategy. Each supports the other and lays the groundwork for the whole to succeed. If you view AI as simply another tool you can apply, you miss out on defining strategies that are only possible because of the capabilities of AI.

It is crucial to connect your overall strategy to your digital strategy and your AI strategy. Each informs the other and enables objectives and courses of action that would not be possible if the different aspects were dealt with in isolation.

At a more mundane level, connecting activities also means determining how to best time and coordinate projects so that you get a positive outcome overall. A typical example of *not* doing this is that as one group within an organization is working to develop capabilities for automated prediction, the software that produces the data that that prediction depends on is already planned to be replaced—the equivalent of pulling the carpet under the feet of the first team.

You want to avoid conversations like the one below:

> "- The new CRM project is well underway—the new system will be up and running by early next year."
>
> "- Will the new CRM be able to supply the relationship data and historical sales data that we need to enable prediction?"
>
> "- I don't know. That wasn't part of the requirements a year ago when the request for proposals went out to vendors."
>
> "…"

Getting a firm grasp on process, data flows, and activities that will influence these is crucial for coordinating a successful AI implementation. It does require effort at the planning stage and it points to the need for wider stakeholder participation so that everyone is aware of how changes will affect activities across the team.

Build Your Roadmap

Once we have a better grasp of where we are, we can start planning out the steps that will take us to where we want to be. At the very highest level I find it useful to consider three broad possibilities. In part, these three approaches can be viewed as stages along the evolution of your AI capabilities, but ultimately they are three streams that you can follow concurrently and at times you can decide to move from one to the other. These three streams are:

- **Hire tools with AI built in.** In this case we are looking for tools with AI capabilities already built in. We don't need to develop anything from scratch. Simply take advantage of what is available.

- **Build solutions with prebuilt AI components.** Here we are developing our own custom solutions, but when it comes to using AI techniques or capabilities, we don't train or develop our own models. Instead we use AI services that are pretrained or in some way prepackaged to give us the functionality we need.
- **Build AI components and infrastructure.** Finally, we can consider building our own AI models to include within wider solutions. This step could be further divided into building AI models using existing techniques or developing new techniques to help us derive models.

We will consider each of the options in more detail in the next sections.

Hire Tools with AI Built In

A very straightforward choice is to "hire" services with AI capabilities built in. This provides an immediate step on the AI evolution ladder without having to dedicate significant resources to build something internally. There are thousands of vendors vying to provide intelligent capabilities to businesses, and taking advantage of this innovation is a great way to see how AI capabilities can enhance your current workflows.

From a practical, implementation perspective it means adding another dimension to your purchasing decisions, whereby you explicitly evaluate the possibilities that a service creates around automation and how those possibilities can address your needs. There are two key questions to be considered:

Is it addressing a real need within the organization? Is it solving a real problem we are facing that would benefit from automation? This seems like an obvious question, but it protects against "checklist purchasing"—where purchase decisions are done by a separate department within an organization, and all that department is looking to do is tick off the "AI capabilities" on their list. As we have already said numerous times in this book, AI techniques vary greatly. In many ways simply specifying "AI capabilities" is about as useful as specifying that a computer program should use a programming language! Instead, we have to address the specific capability or set of capabilities we are looking for with respect to the problem we are trying to solve.

How is the underlying data treated? Will you be able to use the data without that product? Will data produced be able to be fed back in the virtuous cycle we described at the start of this chapter? I believe this is crucial. Going back to the data rating scheme we described earlier, we can judge what type of data will be produced from the system we are hiring and what level of lock-in to a specific vendor this creates. The understanding generated through your own activities is valuable intellectual property that ideally should be closely guarded. In a situation where all competitors are using the same tool, it is the data and process configuration of that tool that can give you a competitive advantage.

All the major vendors offer interesting solutions that allow you to "hire" solutions with AI built-in. For example, the major CRM vendors (Salesforce, Oracle, SAP, Adobe, Microsoft) have all bundled AI capabilities within their CRM tools. Let's briefly consider what Salesforce has done, as it is a real-life example of the type of tactics we are discussing here. The first step was to recognize that offering AI capabilities within their CRM would represent a key competitive advantage and potential differentiator. Then, in order to quickly build up their AI capabilities they went on an acquiring spree, purchasing AI startups that either provided specific functionality such as intelligent meeting management (so tools with AI built in). Then they purchased companies that borough lower-level techniques into the mix so that they can, for example, create a machine learning platform - enabling building with AI. All these startups were eventually combined into a comprehensive solution called Einstein, enabling Salesforce to build new AI techniques and capabilities. Einstein provides features such as accounts insights, leads prioritization, and automated data entry.

Build with AI

The next approach to take is to build solutions using easily accessible AI components. In this case you are not hiring the final functionality with its predefined UI and feature list outright. Instead you are building your own tool (say a dashboard to be able to view potential candidates ranked) and you are using an external AI platform to provide you with the necessary capabilities (e.g., natural language processing).

Once more, all the major technology providers offer easily accessible APIs along these lines. Microsoft's cognitive services include tools to enable Decision, Vision, Speech, Search, and Language. Amazon AWS calls theirs Recommendations, Forecasting, Image and Video Analysis, Advanced Text Analytics, Document Analysis, etc. As you can see just by the names of the services, these are broad capabilities that can be fed with your own data and composed to provide more comprehensive solutions

Of course it's not just about cloud-based services from the big providers. There is a wealth of open source tools such as spaCy for natural language understanding that provide incredible capabilities and require little effort to get started.

The appeal of these ready-made capabilities is that you can plug into your solutions without requiring specialist in-house skills to understand them. Your differentiator, once more, is in how you compose the solution and the quality of the data and overall problem understanding that you bring to the table.

Build AI

The last step is to actually invest in building basic AI internally. This makes sense once you have more clarity about the state of your processes and data and can start building your own models, combining foundational techniques such as neural networks, reasoning, etc. It will mean building out AI skills in-house and requires a bigger investment, but it is also the space that is likely to bring the most interesting returns because this is the area that you can truly differentiate what you do. The balancing act to handle here is ensuring that you are investing in the right direction in building your own AI and are not simply trying to compete with technology behemoths.

AI Everywhere

Developing and defining your own AI strategy, as with any high-level strategic work, is a challenging but ultimately highly rewarding activity. It means that you will need to dig deep into understanding your own motivations and the motivations of your colleagues and organization as a whole. It means looking at how you solve problems and attempting to derive explicit rules and clarity. That process alone is incredibly valuable. It is one thing to look at processes for the purpose of documenting them for other humans and quite another to look at that same process and attempt to describe it to a machine. It forces a level of clarity that at times may even feel uncomfortable or awkward. The outcome, however, will undoubtedly be very valuable.

We've seen that there are quite a few options about how to get started on the journey from hiring AI, building with AI, and building AI. Each has its own advantages and disadvantages and while they may feel like different steps along an evolutionary ladder, they are not mutually exclusive. They can coexist, and you can make different choices for different use cases within your organization. It also means that you can get started quickly and you can start showing the benefits of an AI strategy early on, which in turn will fuel further support and make the next steps easier to sell to the rest of the team.

AI techniques and capabilities will influence every aspect of the workplace. I hope I have demonstrated throughout this book that I am not one to get excited by fads and hype. Given how hyped AI technologies are at this point in time, it may be hard to see through that to what their real impact can be. However, as we argued in the first chapters, the impact of AI will be far-reaching. Comparing it to fire and electricity (as companies such as Microsoft and Google are doing) may sound far-fetched. There is truth in those statements though. Just like fire or electricity, AI has the ability to change how we do everything. If you start from a position that is pragmatic about the challenges but also recognizes the opportunities, you can make a lasting impact on how work is done in your organization. Getting started on an AI strategy may be one of the most significant decisions for the future of your organization.

CHAPTER

12

The Ethics of AI-Powered Applications

Why do we need to talk about ethics in the context of AI-powered applications? Isn't it just like any other piece of software or any other machine? Aren't the rules that are already in place enough to cover us? Is this about machines taking over the world!?

Let's start with the last question. No, this is not about machines taking over the world. There are, undoubtedly, hugely interesting ethical considerations that we will have to tackle if and when we get to the point of having machines that act autonomously with artificial general intelligence. However, as we already discussed in Chapter 2, there is no immediate worry of that happening and even if it somehow did happen, this is not the book that is going to tackle the ethical concerns raised. The issues we are concerned with here are not around machines taking over the world. We are concerned with machines behaving in ways that are not safe, where their actions cannot be explained, and that lead to people being treated unfairly without any way to rectify that.

R. Ashri, *The AI-Powered Workplace*,
https://doi.org/10.1007/978-1-4842-5476-9_12

AI-powered applications merit specific consideration because software that performs automated decision-making is materially different from other types of software. As we discussed in Chapter 3, we are dealing with software that has a certain level of self-direction in how it achieves its goals and potentially autonomy in what goals it generates. Most other software is passive, waiting for us to manipulate it in order for things to happen.

Furthermore, the level of complexity of AI software means that we need to explicitly consider how we will provide explanations for decisions and build those processes in the software itself. At this level of complexity, the path that led to a specific decision can easily be lost. This is especially true of data-driven AI techniques, where we are dealing with hundreds of thousands or millions of intermediate decision points (e.g., neurons in an artificial neural network) all leading to a single outcome.

Therefore, precisely because AI-powered software is not like other software, we have to explicitly address ethical considerations, how they can weave themselves into software, and how we uncover and deal with them. With AI we are not programming specific rules and outcomes. Instead we are developing software with the capability to infer, and based on that inference make choices. Put simplistically, it is software that writes software. We, as the creators, are a step away from the final outcome. Since we are not programming the final outcome, we need to build safeguards to ensure it will be a desirable one.

The Consequences of Automated Decision-making

All of that introductory reasoning may have felt a bit abstract, so let's try and make it more real with a practical example.

A subject that, thankfully, is being discussed increasingly more frequently within technology circles is how to address the huge inequalities that exist within the workplace. Gender, religion, ethnicity and socio-economic status all impact what job you are able to get and how you are treated and compensated once you do get it. The ways this happens are varied, with some being very explicit and some more subtle.

Here is an example of a very explicit type of discrimination that was recounted to me by an engineer living in Paris. He explained how a friend asked the favor to use his address in job applications. When asked why, the friend explained that if he used his own address the job application stood a higher chance of being rejected. The friend lived in a suburb that was considered poor and rife with crime. It turns out that recruiters used the postcode as a signal to determine the applicant's socio-economic status.

Now, assume that those same companies decide that they should build an automated AI-powered tool to help do an initial sift through job applications. As we discussed in previous chapters, the way to do it would be to collect examples of job applications from the past that met the "criteria" and examples of job applications that did not. The AI team will feed all the data into a machine learning algorithm and that algorithm will adjust its weights so as to get the "right" answer. While individual members of the team preparing the tool are not necessarily biased, or looking to codify bias, they will end up introducing bias because the data itself is biased.

The algorithm will eventually latch on to the fact that postcodes carry some weight in decision-making. These algorithms are, after all, explicitly designed to look for features that will enable them to differentiate between different types of data. Somewhere in a neural network, values will be adjusted so that postcodes from economically disadvantaged areas negatively affect the outcome of the application. The bias and inequality have now been codified—not because someone explicitly said it should be so, but because the past behaviors of human beings were used to inform the discovery of a data-driven AI-based reasoning model.

This hypothetical scenario became a very real one for Amazon in 2018. The machine learning team at Amazon had been working on building tools to help with recruitment since 2014. The CV selection tool was using 10 years' worth of data and the team realized that it favored men over women. The algorithm simply codified what it saw in data. The overwhelming proportion of engineers was male. "Gender must play a role," the algorithm deduced. It penalized resumes that included the word "women's," such as "women's chess club captain." It also downgraded graduates of two all-women's colleges.[1] Even if the program could be corrected to compensate for these particular instances, Amazon was too concerned that they would not be able to identify all the ways in which the predictions may be influenced.

You can imagine in how many different scenarios similar biases can be introduced. Using past data to inform decisions about whether someone should get a mortgage or not, what type of health insurance coverage one should have, whether one gets approved for a business loan, or a visa application, and the list goes on. In the workplace, what are the consequences of automating end-of-year bonuses calculations or how remuneration is awarded in general?

Even seemingly less innocuous things can end up codifying and amplifying preexisting patterns of discrimination. In 2017, a video went viral that showed how an automated soap dispenser in a hotel bathroom only worked for lighter

[1] www.reuters.com/article/us-amazon-com-jobs-automation-insight/amazon-scraps-secret-ai-recruiting-tool-that-showed-bias-against-women-idUSKCN1MK08G.

skin tones.[2] The soap dispenser used near-infrared technology to detect hand motion. Since darker skin tones absorb more light, the dispenser didn't work for them. Not enough light was reflected back to activate the soap dispenser. It was not the intention of the designer of the soap dispenser to racially discriminate. But the model of behavior they encoded for this relatively simple decision did not take into account darker skin tone. It was a faulty model, and at no point from inception to actual installation in bathrooms was the consideration made about whether it would work for all skin tones even though it depended on the hand's ability to reflect light.[3] Now, assume you've just made a significant investment in your own organization to improve the workplace, one that included an upgrade of all the physical facilities. To great fanfare the new working space is inaugurated; big words are uttered about inclusion, well-being, and so forth. Colleagues with darker skin tones then realize that the bathrooms will not work for them. Even if people decide to approach this lightly and not feel excluded, that sense of exclusion at some level is inevitable. It reminds them of the wider injustices in everyday life and the lack of diversity and consideration of diversity.

Automated decision-making will encode the bias that is in your data and the diversity that is in your teams. If there is a lot of bias and very little diversity, that will eventually come through in one form or another. As such, you need to explicitly consider these issues. In addition, you need to consider them while appreciating that the solution is not just technological. The solution, as with so many other things, is about tools, processes, and people.

In the next section we will explore some guidelines we can refer to in order to avoid some of these issues.

Guidelines for Ethical AI systems

In order to avoid scenarios such as the ones described previously, we need to ensure that the systems that we build meet certain basic requirements and follow specific guidelines.

The first step is the hardest but the simplest. We need to recognize that this *is* an issue. We need to accept that automated decisions-making systems can encode biases and that it is our responsibility to attempt to counter that bias. In addition, we also have to accept that if we cannot eliminate bias, perhaps the only solution is to eliminate the system itself.

---

[2] www.mic.com/articles/124899/the-reason-this-racist-soap-dispenser-doesn-t-work-on-black-skin.

[3] This, by the way, is also why diversity can help teams design better products. Clearly, at no point from inception to installation of this soap dispenser did a dark-skinned person interact with it. There was likely nobody in the team that designed this to pick up on the potential issue.

That last statement is actually a particularly hard statement for a technologist like me to make. I am an optimist and strongly believe that we need technology in order to overcome some of the huge challenges we are faced with. At the same time, I have to accept that we have reached a level of technological progress that is perhaps out of step with our ability to ensure that technology is safe and fair. In such cases, as much as it feels like a step backward, we have to consider delaying introducing certain technological solutions. Unless we have a high degree of confidence that some basic guidelines are met, that may be our only choice.

Between 2018 and 2019 the European Union tasked a high-level expert group on artificial intelligence with the mission of producing Ethics Guidelines for Trustworthy AI.[4] The resulting framework produced is a viable starting point for anyone considering the introduction of automation in the workplace. We will provide a brief overview of the results here, but it is worth delving into the complete document as well.

Trustworthy AI

Trustworthiness is considered the overarching ambition of these guidelines. In order for AI technologies to really grow, they need to be considered trustworthy and the systems that underpin the monitoring and regulation of AI technologies need to be trusted as well. We already have models of how this can work. It is enough to think of the aviation industry—there is a well-defined set of actors from the manufacturers to the aviation companies, airports, aviation authorities, and so on, backed up by a solid set of rules and regulations. The entire system is designed to ensure that people trust flying. We need to understand, as a society, how we want the analogous AI system to be.

For trustworthy systems to exist, the EU expert group identified three pillars. AI should be lawful, ethical, and robust. We look at each in turn.

Lawful AI

First, AI should be **lawful**. Whatever automated decision-making process is taking place we should ensure that it complies with all relevant legal requirements. This should go without saying. Adherence to laws and regulations is the minimum entry requirement. What specifically needs to be considered is what processes are in place to achieve this. Companies in industries that are not heavily regulated may not be accustomed to questioning the legality of the technical processes they use.

[4] https://ec.europa.eu/digital-single-market/en/news/ethics-guidelines-trustworthy-ai.

Ethical AI

Second, AI should be **ethical**. Ethics is, clearly, a very complex subject and one that cannot be entirely reduced to a set of specific concepts. The expert group grounded their approach to identifying a set of principles for ethical AI through the recognition of some fundamental rights, as set in EU Treaties and international human rights law. These are

- *Respect for human dignity*: Every human being has an intrinsic worth that should be respected and should not be diminished, compromised, or repressed by others, including automated systems. Humans are not objects to be manipulated, exploited, sifted, and sorted. They have an identity and cultural and physical needs that should be acknowledged, respected, and served.
- *Freedom of the individual*: Humans should be free to make decisions for themselves. AI systems should not be looking to manipulate, deceive, coerce, or unjustifiably survey.
- *Respect for democracy, justice, and the rule of law*: In the same way that AI systems should respect individual freedom, they need to respect societal processes that govern how we manage ourselves. For example, AI systems should not act in a way that undermines our ability to have trust in democratic processes and voting systems.
- *Equality, nondiscrimination, and solidarity*: This speaks directly to the need for AI systems to avoid bias. Society, in general, has been woefully inadequate in addressing these issues. It is enough to look at something like equal pay for men and women to admit that such issues cannot be resolved simply by saying people should act with respect for each other and lawfully. As such, it is important that we reiterate the point of equality and solidarity, in addition to the ones mentioned before.
- *Citizens' rights*: In this book we focused on how AI systems can improve life in the workplace. Similarly, AI systems can improve life for all of us as citizens, as we go about interacting with government administration at various levels. Equally, however, AI systems can make those interactions opaque and difficult. Specific consideration needs to be paid to ensure that that does not happen.

Building on these rights, they went on to define four ethical principles, namely:

1. *Respect for human autonomy*: AI systems should not look to manipulate humans in any way that reduces their autonomy. Some aspects, such as an automated system forcing a human being to do something, are "easier" to identify. Things become more challenging when we are designing systems that more subtly influence behavior. Are the goals and purposes of our system transparent, or is it trying to manipulate behavior in order to achieve a goal that is not clearly stated upfront?

2. *Prevention of harm*: Harm in this context is not referring simply to physical harm, which might be easier to pinpoint and justify. This is also referring to mental harm and both individual and collective societal harm. For example, some AI tools can be incredibly resource hungry. The amount of calculations required means that significant amounts of energy are expended[5]. If you were to develop an AI-powered tool that required an inordinate amount of energy, are you considering that cost (that is less obvious) as something that is causing harm? It is not that different from considering what your organization does with respect to energy efficiency in general, and whether that is not only a financially sound thing to do but also an ethical principle of not causing harm to the environment. Obviously, none of these questions have easy answers. The first step is to consider them and have honest discussions about what can be done.

3. *Fairness*: There are no simple answers or a single definition of fairness. Basing our thinking on the rights defined earlier, however, we can say that fairness should be a core principle at the heart of the design on any AI system, since lack of fairness would, at the very least, lead to discrimination. We could also take it a step further and say that AI systems should try to improve fairness and actively work to avoid deceiving people or impair their freedom of choice.

[5] There is an increasing recognition of how energy hungry the entire IT industry is. According to research by the Swedish Royal Institute of Technology the internet uses 10% of the world's electricity. AI techniques only exacerbate energy demands.

4. *Explicability*: If a decision of an automated system is challenged, can we explain why that decision was made? This goes right to the heart of the matter. Without explicability, decisions cannot be challenged and trust will very quickly erode. Is it enough to say that the reason someone was denied a vacation request or a pay rise is because a system trained using data from past years decided that it was not an appropriate course of action, without being able to specifically point to the elements relevant to that person's situation that contributed to that decision?

It is understandable if the sum of all these issues seems like an insurmountable mountain to climb. Do we really need to go into the depths of ethical debates if all we want to build is a more intelligent meeting scheduler for our company? My personal opinion is that we do need to, at the very least, consider the issues. We need to shift our thinking away from considering ethical considerations as a burden or an overhead.

This is about building workplaces that are fairer and more inclusive. Such workplaces also tend to lead to better outputs from everyone, which means a better overall result for an organization. This is not about fairness and inclusivity being better for the bottom line of the company though. It is about whether you consider it a better way to be and act in society.

The more aware we are of the issues and the more questions we pose, the less likely we are to build systems that deal with people unfairly. Even an innocuous meeting scheduler has the capacity to discriminate. It might not take into account the needs of parents or people with disabilities, by consistently scheduling meetings at 9 a.m. or scheduling consecutive meetings in locations that are hard to get to.

There are no easy answers to these questions, and there is constant tension between the different principles. The EU expert group on AI set out a number of high-level requirements to help navigate this space, all leading to more robust AI.

Robust AI

Robust AI refers to our ability to build systems that are safe, secure, and reliable. Let's quickly review some of the key requirements to give a sense of the types of things that we should or could be concerning ourselves with.

- *Human agency and oversight*: We discussed autonomy in Chapter 3 as the ability of a (software) agent to generate its own goals. The limitation on software agency is that it should not hamper the goals of a human, within appropriate context, either directly or indirectly. Oversight, on the other hand, refers to the ability for humans to influence, intervene, and monitor an automated system.
- *Technical robustness and safety*: Planning for when things go wrong and being able to recover or fail gracefully is a key component of any sufficiently complex system, and AI-powered applications are no different. They should be secure, resilient to attacks, and fallback plans should be in place for when things go wrong. In addition, they should be reliable, accurate, and their behavior should be reproducible. Just like any solid engineering system, you need to be able to rely on it to behave consistently.
- *Privacy and data governance*: In this post-Cambridge Analytica[6] world we are all hopefully far more aware of how important robust privacy and data governance are. Because of the reliance of AI capabilities on access to data, it is also a hotly contested issue of debate. With the release of the GDPR regulations in Europe, many said that this would sound the death knell on AI research in the continent. Such regulations hamper access to data, which in turn reduces the speed with which AI research can be done and the final performance of those systems. At the same time, it was heartening to see voices from within large tech companies (e.g., Microsoft's CEO Satya Nadella[7]) accept that GDPR is ultimately a positive thing and invite wider scrutiny. Most recently, Facebook has been proactively asking governments to introduce more regulations (although not everyone is convinced of the motivations behind that). Overall, I think more people are beginning to appreciate that governance is required at all levels, and lack of it will lead to a potentially too strong backlash against technology—a backlash that may prove far more costly than having to adhere to regulations upfront.

[6] www.theguardian.com/uk-news/2019/mar/17/cambridge-analytica-year-on-lesson-in-institutional-failure-christopher-wylie.

[7] www.weforum.org/agenda/2019/01/privacy-is-a-human-right-we-need-a-gdpr-for-the-world-microsoft-ceo/.

- *Accountability for societal and environmental well-being*: Society is coming to the realization that everything that we do has an impact that is not directly visible in our profit and loss statements, and that we carry an ethical responsibility to consider that. In particular, the societal and environmental impact of the systems that we build should no longer be dismissed, and the responsibility for it cannot be offloaded somewhere else. That is one aspect of being accountable, with the other being a much more formal way of tracing accountability and putting in place ways to audit AI-powered applications.

Ethical AI Applications

To build ethical AI applications, the rights, principles, and requirements previously described need to be supported with specific techniques. There is a burgeoning community of researchers and practitioners who are working specifically in this direction.

From a technical perspective there is research toward explainable AI, and methods are being considered to help us marshal the behavior of the immense reasoning machines and neural networks that we are building. There is also much needed interdisciplinary work to get technologists talking more closely with other professions. It's only through a more well-rounded approach that considers all the different dimensions of human existence that we will be able to move forward more confidently.

From a societal perspective, governments (and we as citizens) have to look for the necessary structures to put in place in order to support trustworthy AI. We will need appropriate governance frameworks, regulatory bodies, standardization, certification, codes of conduct, and educational programs.

As we think about how to introduce AI in our workplace, we also play a role and carry a responsibility in this context. The first step is about educating ourselves and becoming more aware of the issues. The second step is about building these considerations into our process and allowing discussions to take place.

It is not an easy process, and it does require specific effort. However, this is the time for us start working toward a future where the impact of the technologies we develop is much more carefully considered. If we do not invest the necessary effort in building trustworthy AI, we risk having to deal with the far more serious aftermath of disillusioned societies and people. The workplace is a large component of our lives. We, hopefully, do not want to build workplaces where people feel surveilled and controlled.

Technology can be liberating as much as it can be disempowering. It can create a more fair and equitable society, but it can also consolidate and amplify injustice. We are already seeing how people feel marginalized by the introduction of wide-scale automation in manufacturing. The broad application of artificial intelligence techniques in every aspect of our lives will be transformative. It is up to us to ensure that that transformation is positive.

An AI-powered workplace can be a happier, more positive, more inclusive, and more equitable workspace. We will not get there, however, without carefully considering the ethical implications of what we are doing. There is no free lunch, even in a fully automated office. We need to put in the extra time and resources required to ensure that we build a better workplace for today and contribute to a better society and a healthier environment for tomorrow.

CHAPTER

13

Epilogue

A Day at Work in 2035

It's 9:15am, Monday morning in London. Leo logs into his company's online collaboration space, AugmentOS. The presence map gives him an overview upon login. Most of the European team is already online. He waves it away. AugmentOS can recognize gestures as well as listen for voice commands. You can also go old-school and type in what you need it to do. Leo's usually not interested in who is online at any given moment, although it's nice to get a quick look. Also, he has to admit that he gets a kick from looking at the beautiful 3-D map with the people indicators lighting up all around the world.

He chuckles as he recalls the days of Slack or Skype and their green presence dots. How rudimentary those interfaces look now and how amazingly more powerful AugmentOS is. Slack and its cohorts were a key part of the transformation that brought workplace tools to where they are now. These days, work without tools like AugmentOS cannot even be conceived of. To think that just 15 years ago only a few tens of thousands of organizations used conversational collaboration environments! How did he get anything done back then?

His attention is drawn to a message that zooms in from the back of the virtual space on his VR/AR headset. He prefers the concentration afforded by VR to the mixed reality the headset also supports. The message must be important if his automated personal assistant let it through. He's been working with this AI for a year now, and it knows he values an interruption-free start more than anything.

R. Ashri, *The AI-Powered Workplace*,
https://doi.org/10.1007/978-1-4842-5476-9_13

It's his boss, Sofia. She lives on the west coast of the United States, so the message is a few hours old. Last night they received an unusually high number of support requests. The automated support system had to route more than 30% of all questions to human operators, and the European team will have to help with the load.

Leo pulls up their data analysis tools and has it run a few tests on all the messages. It looks like most people are frustrated. Lots of different phrases, all describing the same problem, keep popping up. The automated language understanding system is confused, but to Leo it's obvious. It's all about the new feature they released last week. Leo has been training the company's language tools for some time now. He knows their limitations. The words people are using to describe the problems they are facing vary too widely. The automated classifier hasn't been able to figure it out on its own. But now they have more data. They can train it to better handle the way actual customers describe the issue.

The VR space goes dim and a reminder pops up. Leo has been so focused on analyzing data that more than 2 hours have gone by. It's time for a break and catching up with the outside world. Leo likes to disconnect completely on his breaks. Most people use the break as an opportunity to check in with friends in virtual space and tinker with their avatars. He prefers the calmness that switching off affords though. A button on his watch turns off all notifications except a very select list of close family and friends. He goes for a short walk in the park.

After his break, Leo has a virtual meeting with the rest of the NLP team. AugmentOS has been tracking the work Leo did on this problem as well as that of a couple of other people and can provide an accurate synopsis for everyone. They discuss how they can improve future releases to avoid similar issues and what needs to be done to train the system for the current problem. Work assignments and meeting records are automatically generated, and the work will be routed to the next available experts. AugmentOS has an excellent understanding of the skills required and access to a worldwide pool of talent. This means lots of people can pick up where Leo and his team left off.

It's already 12:45 pm. Leo is done for the day. He logs off to go pick up his kids from school. He spends the rest of the day with his family. They need to prepare for their little road trip tomorrow. The kids have been asking to do something fun, so they planned a family day at the beach. An impromptu trip is no problem, as he and his partner only work on Mondays, Wednesdays, and Fridays after all. He has no idea how he managed to work five straight 8-hour days when he started out. The early 2000s were just crazy!

This book talks about how we can use AI techniques in the workplace to delegate aspects of decision-making to machines. It does not discuss what the implications of that will be for society. In Leo's story it sounds like we made a lot of good choices. People get to work less and spend more time with people they care about. However, we do not know how things will evolve.

What we do know is that the way we work and the way we live our lives will change dramatically over the next decades. The defining story will center on how we decide to organize society and apply the opportunities afforded by technological advancement, such as AI, toward solving the great issues of our time.

It is down to each one of us to decide what role we want to play in how we shape and deal with these changes. Will we be passive observers, allowing developments to manipulate us, or active participants contributing to shaping what the overarching goal should be?

I

Index

R. Ashri, *The AI-Powered Workplace*,
https://doi.org/10.1007/978-1-4842-5476-9