I INSIDE COPILOT

Copilot for Microsoft 365

Harness the Power of Generative AI in the Microsoft Apps You Use Every Day

Jess Stratton

Apress®

Inside Copilot

Inside Copilot is designed to teach users to master Copilot, Microsoft's generative AI assistant. Learn prompt engineering and use cases for Copilot in many Microsoft products at beginner, intermediate, and expert levels. Perfect for any professionals who find their schedules packed with repetitive computer tasks, Copilot can automatically generate PowerPoint presentations, draft emails on Outlook, write code on GitHub, and more. Both companies and individuals can learn to utilize Copilot to significantly speed up processes and gain an advantage.

More information about this series at https://www.apress.com/series/17432.

Copilot for Microsoft 365

Harness the Power of Generative AI in the Microsoft Apps You Use Every Day

Jess Stratton

Apress®

Copilot for Microsoft 365: Harness the Power of Generative AI in the Microsoft Apps You Use Every Day

Jess Stratton
Charlestown, RI, USA

ISBN-13 (pbk): 979-8-8688-0446-5 ISBN-13 (electronic): 979-8-8688-0447-2
https://doi.org/10.1007/979-8-8688-0447-2

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Ryan Byrnes
Development Editor: Laura Berendson
Editorial Project Manager: Gryffin Winkler

Cover designed by eStudioCalamar

Distributed to the book trade worldwide by Springer Science+Business Media New York, 1 New York Plaza, Suite 4600, New York, NY 10004-1562, USA. 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 booktranslations@springernature.com; for reprint, paperback, or audio rights, please e-mail bookpermissions@springernature.com.

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 http://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. For more detailed information, please visit https://www.apress.com/gp/services/source-code.

If disposing of this product, please recycle the paper

This book is dedicated to my late father Joseph Spinosa, one of the OG technical writers, and my mother Annthea, who despite being a true scholar of continuous learning will valiantly follow along even though she doesn't use Copilot and that's what makes her amazing.

Also, to my wonderful and brilliant colleagues Marjorie Page and Toby Malina, who introduced me to Copilot and patiently made it possible to learn and explore the brand-new technology together while it was turbulent and still in development.

Finally, to Dr. Joan Peckham, my computer science professor at URI and my OG STEM hero.

Table of Contents

About the Author

Jess Stratton (a.k.a. Nerd Girl Jess) is a tech trainer, productivity expert, and former senior staff instructor with lynda.com/LinkedIn Learning. With over 50 published courses in the LinkedIn Learning library and thousands of hours of live teaching and webinars, she specializes in Microsoft 365, productivity, and the Google Workspace suite of products. Her courses on Excel have over one million views on LinkedIn Learning, and some have even been featured on Delta's in-flight entertainment systems. She has owned and operated a successful brick-and-mortar boutique style computer learning facility and currently has a podcast titled Nerd Girl Jess Untangles Tech. You can find her at nerdgirljess.com.

About the Technical Reviewer

Zewei Song is a developer, support engineer, consultant, and architect who resides in the Chicagoland area. He holds a Ph.D. in control engineering and several professional certifications, such as Certified ScrumMaster, Microsoft Certified Master, Trainer, and Cybersecurity Architect Expert, just to name a few.

Zewei has many years of experience in consulting, proactive support, and technical advisory. He worked for Microsoft for almost 17 years as a senior consultant and field engineer and then joined Discover Financial Services, taking on the role of senior manager in charge of their Microsoft 365 portfolio. Now, Zewei is set to embark on a new journey with Invoke, LLC as a cloud solution architect, where his focus will be on Microsoft Copilot and Purview solutions.

Zewei's aim through work is to be a leader/influencer who helps enterprises maximize their technology investments and enhance end users' productivity. He is happily married and has two kids. He likes to go bicycling and do photography in his spare time.

Acknowledgments

Special thanks to Samara Iodice, Misty Teach-McCloud, and the incredible assets team at my previous workplace for giving me invaluable experience so I could set up my own demo accounts and real-world content for teaching. I do try to ensure that my fictional accounts are not overscheduled though I *am* working on giving them additional hobbies besides coffee breaks and lunchtime walks. Tell Jenny Torres and Leslie Richardson I miss them dearly.

Introduction

When Copilot was announced, it caused a bit of a furor. This new generative artificial intelligence (AI) technology we had been using for about a year to help us be more efficient could now do the same thing in our work environment and directly related to our active work projects. It was – to put it simply – *astounding*. Over the next few months, that furor would turn into more excitement with a side dish of befuddlement. Information posted on the Internet was outdated within weeks. Articles posted without a date were useless. We began speculating in earnest. There wasn't a day in the news without generative AI making *some* sort of headline. Many of you recognized the technology was clearly here to stay yet hadn't had any experience using it. You may still have not yet had a chance to explore what it's all about and what it can do for you.

Fear not, my friend. I'm going to take you through *everything*, and we are starting at the *beginning*. Generative AI is here to stay, and you don't want to be left behind.

You might not know anything and that's OK. I'm glad you're here. You'll be all caught up soon.

The book begins with a few introductory chapters, starting with the history of generative AI. Chapter 1 begins the journey with a quick introduction on some early uses of artificial intelligence in software applications. I'll discuss OpenAI and why it was so integral to the rapid pace of generative AI we are seeing today, what a large language model is and how it's trained, and how to construct a basic and more advanced prompt.

Chapter 2 contains more integral backstory into Microsoft's history of including artificial intelligence into its Office suite of products and how that evolved into Copilot. I'll also deep dive here into the pieces, parts,

and processes of Copilot and how they work together to securely answer questions about your organization's data.

Chapter 3 is the last introductory chapter, containing a short but necessary discussion on responsible AI – what it is, what it entails, and what developers are doing to build trust in their artificial intelligence systems. The goal of this chapter is to help alleviate any fears you may have about using AI and to discuss some risks that come from creating and using it.

Chapter 4 contains the specific details on how you can finally get your hands on Copilot. While this book is very specifically about Copilot for Microsoft 365 in an enterprise environment, I will tell you what *all* the available Copilot options are for the core apps, how to get it, what it will cost, and how to verify your apps have the functionality. It's important to note here that while I do talk about licensing, this is not a book for administrators, so I won't be discussing group policies or enabling Copilot across an organization.

Chapters 5–10 are about how to use Copilot with practical examples in the core apps you use every day – Word, Excel, PowerPoint, Outlook, Teams, and Microsoft Graph-grounded chat. I'll show you each app thoroughly, complete with screenshots and prompts so you can get started using Copilot yourself immediately. Here's where you'll see the scope and magnitude of how much it can help you in your daily professional work life. The goal here is to get you excited and eager to start using it right away in your environment.

Finally, Chapter 11 contains a quick conclusion on the sweeping integration of Copilot into the entire Microsoft ecosystem. Copilot will be integrated into just about every Microsoft application soon, including Windows, GitHub, and Sales, to name a few. Copilot is an all-encompassing brand, and while it's expanding into separate Microsoft products, at the very core will always be the Microsoft 365 products that are used daily by millions of people all over the globe.

Try, try, try. Ask Copilot anything and everything. Don't worry, it will all make sense by the end of this book.

CHAPTER 1

An Introduction to Chat and Generative AI

In 1999, Santa Monica-based company LAUNCH Media Inc. released one of the first of its kind: a new Internet music streaming service. In 2001, LAUNCH Media was purchased by Yahoo! for 12 million dollars to integrate its services and create a wider reaching streaming service, LaunchCast. When it was released, users could listen to a song, rate it on a 1–10 scale, and the service would pick more songs to play. Users could skip a song or rate it, and that choice would be stored with that song to form a complex algorithm that would allow the app to choose the next song for a listener based on similar tastes of other listeners. I used it daily while I worked. It enabled me to be exposed to a whole new library of songs I would never have been exposed to otherwise, and this was an all-day, everyday listening experience for me. As far as I was concerned, this app was a Big Deal.

The rating system later changed to a choice between a 5-star system, or a whopping 0–100 number system, 0 being "never play this again" and 100 being the best song the listener ever heard. The wide range of numbers and the ability to rate musical tastes so specifically allowed the algorithm to really shine. In other words, in all the time I used the app, as an avid song rater, it rarely played me a dud.

J. Stratton, *Copilot for Microsoft 365*, Inside Copilot,
https://doi.org/10.1007/979-8-8688-0447-2_1

In 2005, Pandora launched its consumer version of its similar music streaming service, the same one still used by about 50 million people today. While Pandora still currently allows users to rate a song using a thumbs-up or a thumbs-down system, its primary means of selecting songs based on users' preferences comes from the thousands of hours a handful of musicians and musicologists spent classifying songs at something called the Music Genome Project.

Founded in 2000 by Tim Westergren, the Music Genome Project's mission is to categorize music at its most fundamental levels based on over 450 "genetic" markers that combine to decode its root musical DNA. Using genetic terms is appropriate; after all, it's named after the more well-known Human Genome Project and functions in a similar manner. A collective group of musicians and musicologists have painstakingly cataloged over 60 years of music, spending upwards of 30 minutes on each song, carefully analyzing and categorizing markers such as harmonious melodies, vocal nuances in the singer's voice, whether the primary instrument is acoustic or electric, and even the amount of dissonance in a notable chord. (It was as recently as 2023 that Pandora finally updated the Music Genome Project to a tag-based system, making it easier for modern machine learning models to craft the listening experience.)

The Machines Learn How to Learn

Both LaunchCast and Pandora utilized a pre-artificial intelligence technique that hadn't yet had a name given to it. Today, we call it "supervised learning," a mechanism in which humans are directly responsible for teaching an algorithm how to choose its response in each situation. In LaunchCast, the humans were you, me, and anyone who used the service and rated the songs. In Pandora, the learning initially came from the musicians who cataloged the Music Genome Project and integrated that technology into the app. The *response* was the next song the apps would play.

While it wouldn't be until about 18 years later that "unsupervised learning" would come along, at least definitively titled as such, in the meantime, computers were being taught via programming how to search for patterns by themselves and choose how to respond. Today, chat AI technology has been pre-trained to create a predicted, generated response based on natural language: You enter your prompt in the chat window with what you're looking for, and the AI language model returns with text that is human-like in response.

When we think of machine learning algorithms, it's hard not to think of Facebook and the complex programming it takes to create an algorithm that lets a computer analyze our browsing habits and decide what it thinks we would like to see. However, even if you're not on social media, you've most likely been utilizing the services of an AI algorithm for a while – or at least come into some sort of relationship with it, voluntary or not. If you've ever used Gmail, it uses importance markers to decide what emails are important to you based on signals that include how often we communicate with that person, what specific words we use, and so on.

Google was also using AI in its infancy in the search bar. Aside from creating intelligence to determine common misspellings and instead search for the properly intended word or phrase, Google introduced an AI system called RankBrain in 2015. RankBrain's engine was designed to associate how words relate to concepts, and it used that knowledge to decide what order of importance to show you a site in its search results. This moment in time also marked the introduction of digital voice assistants like Apple's Siri, Amazon's Alexa devices, and Microsoft's Cortana, capable of interacting with you using natural language to perform basic tasks.

If YouTube has ever recommended a video to you or if you browse TikTok even remotely regularly, you've taken advantage of a machine-learned algorithm that has made a choice for you based on data from past or similar transactions. Spell check and autocorrect are all examples of programming that has been trained to anticipate what you're looking for to

make your life easier, and that same autocorrect technology has been built with the assistance and reviews of linguists and computational analysts to create the algorithms. And, if you've ever cursed at a poorly phrased autocorrect, you've already learned that it's not perfect, doesn't always get it right, and absolutely needs monitoring for quality and accuracy. I'm sure you check your junk or spam filter for mail that inadvertently made its way in there incorrectly. Over time, you've no doubt learned, as I have, that monitoring, correcting, and adapting are all parts of the ways we use these tools. When you use AI to generate text, your responsibilities should be the same – monitoring, correcting, and adapting. If it sounds like a lot to worry about, don't worry – soon it will be a natural part of the way we all use the tool.

Even the customer service industry has been using something called *sentiment analysis* early on to detect the overall public opinion on a company's image. The collective posts from X (previously called tweets from Twitter) using a company's mentions or hashtags would be analyzed for words and phrases that indicated a customer had a positive opinion on the company *(great, fantastic, awesome)* versus negative words *(useless, poor, terrible, slow)*, indicating unhappiness and a negative opinion of the company.

As an avid tech trainer (and of course as a user), it's been amazing to watch the evolution of artificial intelligence to the benefit of us, the end users. Autocorrect once only changed words that *everyone* commonly misspelled. Now, as the computing, processing, and Internet speed of each personal device has grown, so has the sophistication. For example, autocorrect now also changes words that are meaningful to you, in addition to the collective masses, and can even change and adjust spelling and grammar for context *while you're still typing the sentence*. Incredible.

I can give you a very clear picture of how rapidly the artificial intelligence industry is evolving by summing it up in one sentence: ChatGPT, which is now a household name and has single handedly changed the way we work and think, was only launched in November

of 2022. Sounds crazy, right? It's absolutely bonkers how quickly the technology has evolved and continues to evolve. Here's the thing: It's impossible to talk about any AI, Copilot included, without making sure you have a proper introduction to OpenAI.

OpenAI Is Created… and Begins Creating

While ChatGPT was launched to the masses in late 2022 and the AI technology would grow to be used by 180 million people as part of their daily lives, the product itself was developed by OpenAI, which was founded in 2015 as an artificial intelligence research organization. The founders list consisted of computer scientists and entrepreneurs, among them Trevor Blackwell, Greg Brockman, Ilya Sutskever, Andrej Karpathy, Vicki Cheung, Durk Kingma, Jessica Livingston, Pamela Vagata, John Schulman, Sam Altman, Elon Musk, and Wojciech Zaremba (Musk later left to focus on Tesla rather than continue a possible conflict of interest with OpenAI development).

In 2018, OpenAI released a report introducing the world to the concept of a "Generative Pre-trained Transformer" (that's where the GPT acronym comes from, most commonly associated with ChatGPT), a method of deep learning in which large language models (LLMs) are trained in an unsupervised manner to predict the next word in a sentence to generate text that *sounds* human.

Let's back up a bit.

Language Models Become… Larger

You'll hear the phrase *large language model* thrown around a lot when learning about generative AI. A large language model is an algorithm, a form of natural language processing that is based on probability and

pattern recognition. In other words, the model is given partial sentences and is then taught how to predict via pattern matching, mathematics, and statistics what most likely should come next to complete the sentence.

Take this sentence for example:

I want to go to the _____.

There's plenty of words that *could* come next here, but there's also a definite set of words that could *not* go next, as it would cause the sentence to make no sense grammatically or practically. The words italicized below can't possibly work, so their candidacy as the next possible word can be eliminated.

I want to go to the _____.

store

car

museum

red

sad

jumping

Once the LLM has used this prediction method to construct the response, it's then taught to continuously transform itself, spreading into longer sentences, paragraphs, and continuing the response to even greater lengths. Given the large scope of the model's ability to predict, transform, summarize, and continuously generate content along with the vast bank of data it's been trained on, it's appropriately titled as a *large* language model.

What are those datasets the large language models are pre-trained on? Content, lots of content, most of which is found on the Internet: web pages, books, articles, journals, etc. Due to the nature of having to pre-train these language models, the various versions of ChatGPT are limited to answers that were acquired during the respective training scope of that particular

version. For example, earlier versions of ChatGPT were limited to a result set dating September 2021 and before. In other words, previously if you had asked ChatGPT what movies a particular actor or actress has been in, it would have only returned those movies before September 2021, even if they've been in movies since then.

This very important backstory does have a purpose, I promise. It all comes back to why you're here, and that's to learn how to use Copilot. And again, it all comes back to OpenAI. Microsoft invested 1 billion toward OpenAI in 2019 and then an additional 10 billion in 2023. A large majority of these funds went toward computing power – the dominant currency in AI. Regardless, the investment opened the doors to integrating AI into Bing in the Microsoft Edge browser based on ChatGPT's technology. Then called Bing Chat (now simply called Copilot), users could harness the power of chat AI right in the browser itself, asking questions about content on any web page. While not an issue anymore, at the time, this was able to eliminate that pesky 2021 date scope barrier.

You may have previously used ChatGPT before or tried it in some way or situation. You may never have typed anything into an AI chat window before, or even *seen* one. Either scenario is perfectly fine for this book – I'm happy you're here, and even if you've chatted with AI before, it's always worth it to have at least a little refresher before we move onto using these scenarios in the Microsoft apps with Copilot, and this foundation can help you make better prompts – or, at the very least, gain an appreciation for the incredible amount of computation that occurs to make a machine sound human.

In fact, even though I started this book's history lesson starting at 1999 with a music streaming service, if you wanted to know when the *first* attempt at conversational AI was pioneered, we would need to go even further back to 1966 when a computer scientist at MIT named Joseph Weizenbaum created the first chatbot named ELIZA. The most popular use of ELIZA was taking the user through a back-and-forth conversation like the way a psychiatrist might interact with a patient, the user being

the patient. While ELIZA was preprogrammed with canned responses and was not sophisticated enough to be used as a real tool, its primary purpose was to explore how we as humans interact with machines using natural language. ELIZA was, however, sophisticated enough to be capable of attempting the Turing test, Alan Turing's methodology that we still use to this day to determine whether a machine or computer system is impersonating a human being or not. (Fun fact: As recent as October 2023, ELIZA beat ChatGPT-3.5 in a Turing test performed by a UC San Diego study on artificial intelligence.)

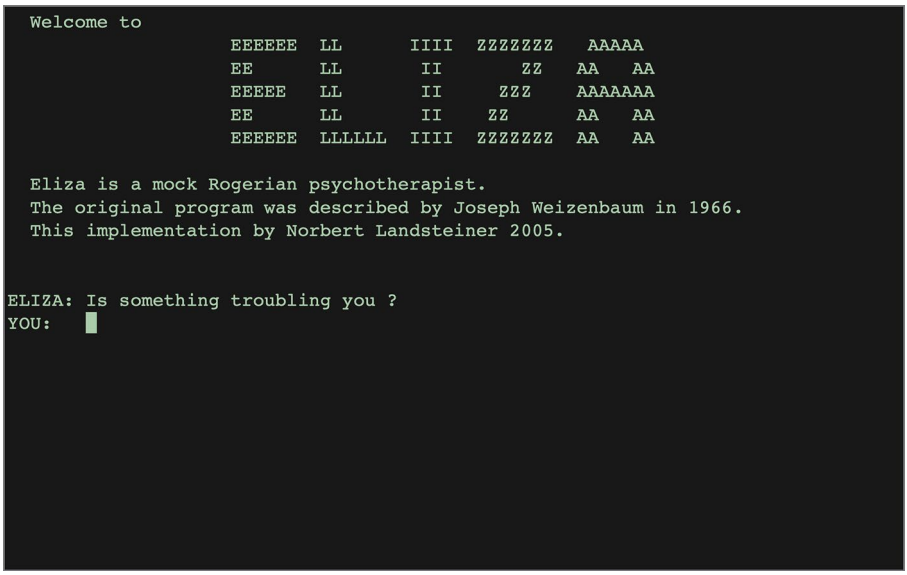

Figure 1-1. *ELIZA is still available in its original form online through emulators*

Another notable AI supercomputer capable of chatting in natural language is IBM Watson. Watson was developed to be a question-answering natural language system and made its debut in 2011 competing against humans on the television show *Jeopardy!*. Watson famously beat

former champions Ken Jennings and Brad Rutter in one such match. Its winnings were donated to charity, and the world was given its first glimpse how a natural language chatbot can serve a *real* purpose – solving problems, sharing knowledge, and answering questions.

Chatting with a Machine Is Like Chatting with a Friend

Today's AI chat capabilities typically work in the same convenient, instant way you'd chat with a friend, or a customer service chat window. You can type your question into the chat window or ask the chat AI to generate something for you. All that computing power that previously examined the knowledge of those large datasets and transformed them into tokens that can recognize patterns then culminates into a response that sounds, seems, and feels like you're talking to another human being. And unlike a search engine, which starts over with a new search every time, you can continue your chat and retain the existing context of the conversation to refine the output with the chat AI, adding more context until you specifically start a new chat. You don't have to restate the problem every time; the context remains in that chat thread.

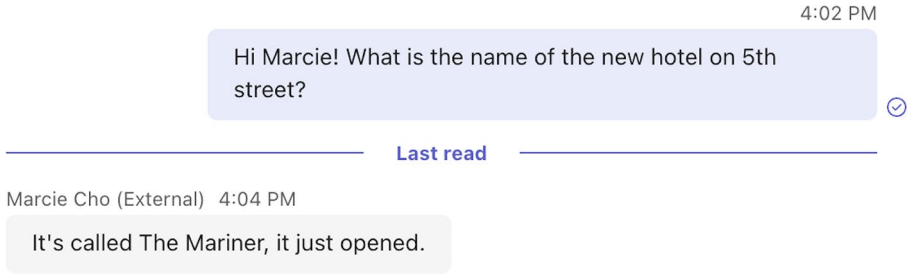

Figure 1-2. *A sample conversion via Teams*