INTERPRETATION ERRORS: EXTRACTING FUNCTIONALITY FROM GENERATIVE MODELS OF LANGUAGE BY UNDERSTANDING THEM BETTER

ARI HOLTZMAN

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington
2023

Reading Committee:
Luke Zettlemoyer, Chair
Hannaneh Hajishirzi
Mirella Lapata

Program Authorized to Offer Degree:
Paul G. Allen School of Computer Science & Engineering

University of Washington

ABSTRACT

INTERPRETATION ERRORS: EXTRACTING FUNCTIONALITY FROM GENERATIVE MODELS OF LANGUAGE BY UNDERSTANDING THEM BETTER

Ari Holtzman

Chair of the Supervisory Committee:
Professor Luke Zettlemoyer
Paul G. Allen School of Computer Science & Engineering

The rise of large language models as the workhorse of NLP, and the continuous release of better models (OpenAI, 2023; Pichai, 2023; Schulman et al., 2022, *inter alia*) has created a strange situation: we have models that are more powerful language generators than ever before, but since we did not design them for a specific purpose we struggle to understand how they should be used or what their idiosyncracies are.

This dissertation describes three empirical projects that sought to characterize the underlying behavior of language models and, importantly, to make them more reliable tools for generating and selecting text where this behavior does not match up with the tasks we would like models to complete. Each project attempts to understand what language models and accompanying inference methods *currently* optimize for, to characterize the gap between that and the true objective of a potential user, and to close it with some new inference method. An emergent theme through these works is that models are already doing what we trained them to do quite well—and it is often the experimenters and practitioners who misunderstand precisely what we trained models to do in the first place. We conclude with a conceptual analysis of how we should study generative models going forward—as models keep improving and new, unanticipated uses and misuses become ever more available.

The first half of this dissertation concerns two works, **Neural Text Degeneration** and **Surface Form Competition**—two failure modes of generative models that occur when probability is viewed as equivalent to "correctness" in text generation and multiple choice scenarios, respectively. For these works we describe the resultant issues, and propose inference methods that largely alleviate them.

The second half of this dissertation goes deeper into the question of how generative models of language capture the communicative goals that humans are optimizing: first with **Learning to Write**, operationalizing communicative goals into auxiliary search objectives for text decoding, and then with **Generative Models as a Complex Systems Science**, which presents a framework to think about the study of generative models as NLP shifts to analyzing systems that are often infeasible to replicate.

How does a model that is predicting the distribution of next tokens understand—and fail to understand—the structure of an essay? This is precisely the kind of question we must face head-on in the new science of generative models.

To Showbiz, baby.

ACKNOWLEDGMENTS

Perhaps it's cliché, but I feel the need to preface my acknowledgements by saying there's simply no way I can acknowledge everyone who has had a significant effect on me. I live my life as a conversation with people, and so many people have added something meaningful to the discussion. I'm sorry to those I missed, as I know I inevitably have.

I have chosen to only list the people within educational institutions, as I began to write a more expansive list and it became completely overwhelming. But I have decided to extend this list of people who created my education back to the beginning, down to my Pre-K teacher who helped me figure out how to take feedback better. My education is one long journey, and I believe the parts before my matriculation at the University of Washington were just the prequel to my doctorate. I have always had a difficult relationship with formal education, and I want to thank the people who made it all worthwhile.

I would like to begin by acknowledging the massive effect Luke Zettlemoyer, my advisor, has had on me. You gave me space to think, an infinitely patient ear, and a perspective that is more thoughtful and gracefully stated than any I have encountered. I cannot say how thankful I am for the environment you created, and for the specific ways you gently nudged me towards being a better version of what I could be as a researcher, being open and considerate about the places I needed to grow past weaknesses as well the places I needed to double-down on my idiosyncrasies and learn to wield them properly.

In a similar vein, I would like to thank Yejin Choi—who taught me style, theory of mind, and what research is made of. I treasure our conversations.

Mirella Lapata: Our discussions in the Summer of 2017 were a formative part of how I came to conceptualize the role I aspired to in academia. Thank you for those discussions and so much more.

Sam Bowman: The way you are capable of articulating the holes in the epistemology of the current research space is something that I admire and strive to do myself. Your accompanying ability to talk to anyone in a civil way that steelman's their side of the issue is something I know I can never replicate, but which I am profoundly moved by. The research community is so much better because you are in it.

Omer Levy: At a critical stage in my PhD, it is your analytical mindset that helped me realize what I wanted out of research. I feel like I become a better researcher with every conversation we have, not just in terms of other peoples' standards, but in terms of the kind of doing the kind of research I know I'll be excited to look back on decades from now.

Mike Lewis: Through your work and our conversations, you have been a role-model to me in thinking about where a field is going before it has the chance to realize it's there, and setting down foundations to make that new thing possible.

Jan Buys: You served as a big brother, showed me the ropes to being a PhD student, to understanding what a paper is, and so much more. You were always very welcoming to my slightly off-kilter perspective, and taught me how to develop the nuance I needed to make it work for me.

Yonatan Bisk: You were a mentor I could come to when I was figuring out so many of the ways in which the research world doesn't make sense to me. Many of the things I was discovering about academia were disappointing, but you were there to discuss how we can push it to become something better. I don't think I could have believed in research while simultaneously learning to navigate the research world without you.

Jesse Thomason: I don't know of anyone I met in my PhD who made me laugh so hard, while pushing me to be ever more nuanced about what on earth I meant by "language" and "communication".

Eunsol Choi: You were one of the first people in grad school I felt I could be honestly myself around, in an institutional culture it took me years to acclimate to. Thank you.

Dallas Card: You are so open to discussion, and the times when we talk and simply take the conversation wherever it goes are some of the times I feel the most at home. You are paragon of true intellectual curiosity, and you have introduced me to so many of the ideas and media artifacts that have become reference points in my thinking. I don't know how to express how much it's meant to be able to share my intellectual dreams with you and to discuss your ideas and perspectives as we predict, argue, and wonder at the passage of history.

Aaron Jaech: Thank you for being the chillest friend I made in the PhD, amiable and welcoming, always down to talk about research or anything else. Your matter-of-factness on the details and challenges of life make it so easy to just talk about what's on my mind, and your insights always reward me for doing so. Every time we meet I regret not having seen you for too long.

Swabha Swayamdipta: You were an inclusive, guiding, and instructional force in my early grad school life, when that was something I badly needed.

Antoine Bosselut: You taught me a great deal about how to express research ideas and results, how to have conversations with academics that would actually go somewhere, and how to think about the research world as it continually changes.

Max Forbes: If it wasn't for you, I think I would have found one excuse or another to drop out in the first few years of grad school. You kept me in and more importantly you made me feel heard when I needed it. Our conversations about what we were doing here gave me a space to figure out what I wanted research to be and what I had to offer academia as a community.

Julian Michael: I feel like you are always embarrassed by my compliments, so I will only say that it means a great deal to me that you spent the time you did taking my perspective seriously. We shall always disagree—that's where the fun is, of course—but you never belittled me for taking on a viewpoint that probably doesn't quite make sense in your framework, while simultaneously enriching my understanding of myself. You, more than anyone on this list, made me want to try harder and do better.

Jack Hessel: You are someone who I feel like I can be as optimistic and cynical as I would like to be, without it feeling like a contradiction. When we talk I feel I can express myself more clearly, and laugh at the world more easily, both of which I truly relish.

Fatemeh Mireshghallah (Niloofar): You are a kindered spirit—in goals, in struggles, in aesthetics—and meeting you at the very end of the PhD preempted and swerved me away from a feeling of lostness I was beginning to feel seep in.

Xuan Luo: You are a friend who understands the trials that the research world puts us through, the conflict between what kind of person you are being pushed to become and who you are right now. Our conversations helped me work out this struggle in a way that I cherish.

MC (Maxwell Christian) Horton: The beverage king, and a big inspiration to me living my life happier and healthier. Thanks for role-modeling the good life in a way I was often too scared to conceive of, while being an A-tier researcher.

Eunice Jun: Our discussions pushed me to define what I actually wanted and believed about my work and philosophy in a way that no one in grad school had pushed me to do before. Thank you for helping me become a me I would like to be.

Rowan Zellers: Every time we talk, you present a point that I never would have thought of, some regularity in these magical new machines we've been exposed to that it seems only you could have come-up with, because you're sensitive not just to what models are supposed to do, but what they actually do. You've always inspired me to try to keep-up with your pace, and I'm incredibly grateful for it.

Kay (Liyiming) Ke: You are the person who changed my mind on the most different things, period. In addition to many other wonderful things, my perspective simply would not be what it is today (and this document would not be possible) without the conversations you took the time to have with me.

Peter West: You saved me from my own mental mazes when I needed saving and you were never afraid to roast me. Okay, you were afraid sometimes. And for that, I roast you here.

Tim Dettmers: It would be *understating* things to say you saved my PhD. Your support, our intense discussions, and our friendship has been a load-bearing pillar in my life for five crazy years.

Gabriel Ilharco: Our discussions on science, on how to live one's life, and on the nature of the changing communities we find ourselves in grounded me and helped me grow.

Sewon Min: You may be the person with the highest value of
(how hard I find it to convince you of my ideas) × (how convinced I am by your arguments),
which is very special to me because I try to seek people who I can sharpen my thinking against. You are a valued colleague and a true friend.

Zhijing Jin: From the moment we had the chance to talk one-on-one I knew you were a live player, someone in the game to change the game. Thank you for being a fellow traveler who I can commiserate, plan, and build with.

Ian Magnusson: We met because of your inquisitiveness, a part of you I greatly esteem. I don't know if I talk to anyone else who thinks so deeply about their place in the world as a researcher and what they want that to mean. Thank you for being an insightful colleague, a dear friend, and someone who pushes me to reflect more deeply.

Kaj Bostrom: I am forever jealous of your ability to go up and down the ladder of abstraction so deftly, and to do so in a way that doesn't exclude people like me who have trouble keeping up. Our conversations have helped me go farther in my thinking.

Ge Gao: You pushed me to think more deeply about abstraction, about who I am as an academic, and about the importance of community in my research life. You are one of the most lucid thinkers I know, you have helped sharpen my thinking on countless occasions, and supported me through my greatest moments of self-doubt. You are a person of incredible dynamic range.

Jared Moore: You are bursting with intellectual energy and a joy to talk to. Thank you for helping me avoid burn-out with your sincerity and willingness to call me out on my silliness. You are the definition of a gentleman and a scholar.

Rosario Scalise: I feel profoundly relaxed around you. Being able to be honest with someone facing struggles similar enough to my own to be relevant and different enough to force us to articulate our circumstances has meant the world to me.

Aaron Walsman: You were the first person I met in grad school who was trying to wrestle with things that were too big to name yet. That made me braver and made me want to try harder to articulate what was worth studying that I couldn't quite put my finger on yet.

Zoey Chen: Thank you for being a person who I could share my struggles with and felt comfortable sharing their struggles with me! You helped me feel okay about having my struggles in the first place, which was often a difficult thing for me to accept.

Katie Stasaski: You helped me navigate academia when I was just beginning to understand what on earth it was, with kindness and finesse.

Margaret Li: Our discussions about what it means to be in grad school, academia, research, ML, and just this time in our lives helped me think through what I wanted out of my career and out of my life. That wouldn't have been possible without your openness to discuss who people really are on the inside. Thank you for sharing that with me.

My Alexa prize team, Hao Fang, Hao Cheng, Maarten Sap, Elizabeth Clark: we DID it. We had fun doing it, too. This was such a formational part of my early PhD life and I'm so glad I got to do it with you all.

There are so many other mentors who have had an influence on me too: Vered Shwartz, Rik Koncel-Kedziorski, Noah Smith, Hannah Rashkin, Mari Ostendorf, Marjan Ghazvininejad, Yannis Konstas, Hannaneh Hajishirzi, Chandra Bhagavatula, Ronan Le Bras, Nicholas Lourie, and Leila Zilles.

To all the folks from the XLab, my time there was so incredibly special: Chloé Kiddon, Saadia Gabriel, Lianhui Qin, Xiujun Li, Ximing (Gloria) Lu, Alisa Liu, and Melanie Sclar.

My ZLabmates: Daniel Fried, Mandar Joshi, Luheng He, Weijia Shi, Terra Blevins, Mark Yatskar, Artidoro Pagnoni, Suchin Gururangan, Victoria Lin, Bhargavi Paranjape, Kenton Lee, Victor Zhong, Hila Gonen, Christopher Clark, and Srini Iyer, you all helped make my grad school a profound experience that changed my perspective in more ways than I can count.

My first year roommates: Bobby Baraldi, Adam Elder, Walter Cai, who made grad school make sense in a way that is hard to describe.

To the folks I've helped mentor: Pooja Sethi, Yao Dou, Leo (Li) Du, Brendan King, Emily Louise Allaway, and Marcella Cindy Prasetio it was such a joy to see you grow! Thank you for giving me the opportunity to grow as a mentor, I'm proud to see you thrive.

There are so many more people who helped me through the PhD: Sebastin Santy, Kelvin Luu, Samuel Ainsworth, Esther Jang, Phoebe Mulcaire, Sofia Serrano, Krishna Pillutla, DJ (Dhruv Jain), Willie Agnew, Roy Or-El, Leah Perlmutter, Xiang Lisa Li, Yanai Elazar, Ofir Press, Morelle Arian, Valentina Pyatkin and many more!

I would additionally like to thank the people who made school possible for me. I have always struggled being in school. In general, it's not a place that allows me to think the way I would like to or discuss things the way I feel would be most clear or productive. I got here because there are many, many people who made school a place where I could learn and grow, and I list here a small fraction that have made incredibly strong contributions to my education below.

Thomas Wies: Thank you for seeing my curiosity and excitement and getting me into research—I had no idea what I was doing or that it would end up being my entire world.

Peter Rajsingh: You set a tone for what it means to be educating one's self that gave me permission and guidance to become the exploratory thinker I wanted to be.

Dmitry Zakharov: The way you approached math reminded me that I had been avoiding going deep into math again because I had forgotten how to make it fun.

Scott Korb: I have met few people with the sensitivity you have to what communication is. The fact that you created a space to think clearly about it together is one of the things I treasure most about my experience at NYU.

David Schalkwyk: I was always fascinated by language, but you gave me the words to explore that fascination, which eventually became the inquiry I dedicate my life to.

Drew C Youngren: Mathematics is often taught more as a competition than as an exploration. Your thoughtfully guided, tastefully participation based, "mathematics as making things clear" pedagogy is the style I would most like to emulate when teaching technical subjects.

Ethan Harkness: Your class "Play and Games in Early China" had a profound effect on me. It convinced me, past doubts I had before, that I wanted to be a person who brings together disparate ways of viewing things into a new architecture. That thing I have decided on is the dynamics of human communication, and the metaphor of a game and your thoughtful attitude towards thinking about games and what they mean in context has always stuck with me.

Lena Feinman: you taught me how to argue about mathematics properly.

Peter Brodie: Our aesthetics don't match, but you may well be the first person who was willing to argue with me on the level of aesthetics, to admit that aesthetics are what make meaning and that they are important and worth contesting. I owe you a great debt for that.

John Schafer: At some fundamental level, you take people who have basic civility and consideration with a seriousness that became a model for how I seek to treat people.

Carla Pugliese: Our advocacy was a place where people genuinely talked about what was on their minds, and I think you have to be very clever, very thoughtful, and a little bit of hero to make that happen. You are those things.

Charles Hanson: You showed me I was a loser, and so I made progress.

Eugenia McCauley: After studying chemistry with you, I always have a tinge of regret for not dedicating my life to its study. I still use the way we learned to think about the properties of complex systems in your class to think about the world around me, and I am very grateful to you for opening up that door.

Michael Thibodeaux: You always engaged with my ideas, however half-baked they were. I wanted to study something I didn't know how to describe. It would take me another decade to begin to describe it, but you helped me incubate it in a way I am deeply grateful for.

Richard Steinberg: I always liked programming, but you taught me what Computer Science really is, and I now dedicate my life to trying to make the language of Computer Science describe a frontier that still remains somewhat elusive: the mechanics of human communication.

James Dann: You facilitated a space where science could be what I wanted it to be—exploratory, challenging, exciting. You are one of the kindest teachers I have ever had. You helped me believe that by going in deep enough on what I wanted to study I could make something out of it—even as I failed again and again to quite find that thing. The space you created to fail is incredibly important, an underrated need in learning how to do exploratory research, and the foundation of my mindset in my current research. I am overwhelmingly grateful.

Deb Jensen: I loved physics long before I took your class, but you taught me through demonstration what physics was actually about: observing regularity in the universe and attempting to characterize it. I still think about the methods of inquiry you taught us all the time.

Rachel Chou, Jackie Arreaga, Rebecca Akers, and Barbara Callaghan: Thank you for teaching me to go beyond high school mathematics.

Mark Newton: It's hard to express how much World Religions, your directorship of the Elephant Man, and your presence as a mentor—as someone who helped support me through an institution that wasn't always charitable to my perspective—meant to me. It's not just that you helped me survive; you are someone who helped me become more of who I am, by showing me places where I had been scared to grow.

Cindy Lapolla: You were incredibly patient and welcoming in helping me adjust to an environment that I had so much trouble making sense of. You also helped me make sense of what I wanted out of learning in a way few engaged with.

David Susman: David, how do I acknowledge you in a few sentences? You have affected my ability to express myself more than perhaps anyone on this list. You have been a guide, a mentor, a friend, a role-model, a reference point, a producer of beautiful writing, an amazing and kind human being. Thank you.

Heather Roark Nodelman: You are a truly kind person, open and communicative. You showed me that the heart of learning languages was really learning how to communicate, and developing the nuances of communication as they become useful to get across one's ideas and feelings.

Camille Geraci: You made a space where I could feel comfortable discovering what art could be to me, instead of forcing me to play to what other people felt could be justified as art.

Name Unknown: Despite the briefness of our interactions, you were one of the closest things I had to a role-model, because you had an incredibly observant eye for people and you showed me the kind of role I could fit into if I leaned into that way of life.

Richard Butterfield: Most of what I learned about rhetoric was discovered by taking the pointers you gave us in a few short lessons in elementary school and using them as tools to explore the world and the hearts of other people.

Azmi Mamis: I had never developed an exercise routine or even a basic framework for thinking about physical activity till your P.E. classes. It would take me another decade and a half to get there, but you taught me to take my physical existence seriously. When I finally did develop a way of being an embodied person, I relied heavily on lessons you put effort and thought into teaching me.

Stephen Lessard: You are the person who taught me what context truly is. Not some banal extra detail that naïvely frames information, but a breathing conceptual ecosystem that causes a continuous shift in perspective. A living organism whose rhythm of life must be studied in order to understand the nature of knowledge.

Liza Raynal: You taught me about poetry, prose, prosody, rhetoric, expression, persuasion, and conviction—and you did all this by showing rather than telling.

Jim Munzenrider: You are the person who taught me that music is a language and showed me the way into its idioms, affordances, and ceremonies, which are so much more vital and hard to express than its grammar. You are also just a cool dude.

Dan Bennett: To be frank, I disagreed with you about almost everything. But you were really quite the gentleman about it, and in the process of discussing it, you taught me a lot and made sure I didn't get lost in a system that would have eaten me alive.

Tracie Mastronicola: 6th grade physics was an awakening to the connection between experimentation and theory that ended-up being a life long fascination for me. Your excitement for it and your willingness to always, unfailingly be present as a human being—and not just the prescribed role of a Teacher—had a huge influence on me.

Peter Koehler: You demonstrated the kind of open inquiry that the process of doing mathematics can be. I wasn't quite ready for that lesson at the time, but it left an impression on me, one which I kept in my heart forever.

Lorri Hamilton Durbin: When we first met, I had little respect for the need for an authority figure to make things run smoothly, especially in education. You demonstrated how much one can bring the best out of an educational institution by leading well, and I have often thought back on your style of leadership since.

Wendy Donner: You were the first person who tried to convince me that I had potential that I wasn't making use of properly. I didn't believe you at the time. But I remembered what you said, what you suggested I pursue, what skills you thought I needed to hone, what perspectives you thought I was missing, and when I was finally ready I used those. Thank you for being the first person to truly call me out.

Suzanne Geller: I think you were the first teacher I felt genuinely comfortable expressing my emotions to, and you helped me make that a bigger part of my life from then on.

Janice Toben: I always disagreed with you about the fundamentals about how feelings worked, and you taught me—more than anyone else—how feelings worked.

Tricia Yao: I eventually came to love writing, but initially it was very, very hard for me to accept what other people wanted out of my writing. You were thoughtful and kind in taking me where I was at and showing me how that could be made intelligible to others. Thank you for being the first person to help me figure out how to make writing a vehicle.

Grant Ditzler: I don't think I ever had another art teacher who took discussing art with me as seriously as you did. The way you would refer to unintended negative space as "holidays", the way you would critique our lack of buying into the premise of the exercise when we would play games like Exquisite Corpse, the way you would debate with me whether I was able to get across what I intended in art or whether I was simply factoring in my lack of skill to lower my standards. I still think about our conversations 23 years later and I cherish them.

Andrew Salverda: You are the first teacher in my life who took me truly seriously and tried to explain why the things that frustrated me about the system I lived in were actually levers I should make use of if I wanted to be happy, to deftly regulate my life in a complex environment. I really can't overstate how much our lunchtime discussions in 2000-2001 meant to me then, and how much they still mean to me now.

Diane Rosenberg: You taught me more about the political economy of social interactions in a few conversations than perhaps anybody else ever managed to.

Marky: You were kind to me, and you recognized that I was sensitive to things, and nudged me on that bit by bit. That sensitivity has been difficult to navigate for my entire life, but your sensitivity toward it helped me learn how to grow with it.

And Menachem, who taught me everything else.

CONTENTS

1 INTRODUCTION

1.1 INVENTED MODELS, DISCOVERED USES

Natural Language Processing (NLP) in the 21st century has been increasingly focused on neural language models, first simple feed-forward neural networks (Bengio et al., 2003), then RNNs (especially LSTMs) (Peters et al., 2018; Sutskever, Vinyals, and Le, 2014), and more recently Transformers (Radford et al., 2018a; Vaswani et al., 2017). These models, applied via finetuning (Radford et al., 2018a), prompting (2019b), and more recently dialogue via instruction tuning (Mishra et al., 2022) and Reinforcement Learning from Human Feedback (RLHF) (Ziegler et al., 2019), have been incredibly performant on many traditional tasks, even reaching super-human levels in some cases (Goyal, Li, and Durrett, 2022). Most Natural Language Understanding (NLU) and Natural Language Generation (NLG) leaderboards are dominated by finetuned or prompted language models (BigBench, 2021; Gehrmann et al., 2021; Wang et al., 2019, *inter alia*).

Increasingly, models are used for purposes they were never designed for. Instead, models are first trained for very general objectives, such as predicting a document one word at a time, and then desired behavior is coaxed out of them via a combination of prompting, finetuning, and reinforcement learning. This situation, in which we often do not have a precise formalization of what function models compute, results in the need to understand models better in order to design such inference methods properly. It equally leads to the question: how can we better formalize what we want models to compute? This problem has often been side-stepped by fixing obvious model errors, without fully defining what success conditions for a model might be.

Understanding language models better in order to design better methods for using them is the subject of this dissertation. We analyze language models through their relationship to their component parts, to their training objectives, to the human communication they are mimicking, and to simplified models of what they are doing. We focus on the problem of *interpretation errors*: instances where the conventional assumptions about what model behavior is optimizing do not hold and we must reconsider what models are doing and how to best make use of this discovered functionality.

This highlights one of the most interesting aspects of modern NLP: while we invent new models that are ever-more capable, we generally do not understand what they will be capable of before we experiment with them. This makes the study of generative models more akin to a natural science, like astronomy or meteorology, where we attempt to discover what is already there, rather than a traditional engineering science of building systems capable of specific tasks.

In the rest of this chapter, we describe some of the most common interpretation errors that have been made regarding what language models have been optimized to do, with a short preview of how the rest of this dissertation addresses these issues.

1.2 ARE LANGUAGE MODELS SIMULATING WRITERS?

The most common kind of language model used for NLG tasks is the *causal language model*, in which the tokens that compose a document are predicted from beginning to end, one by one. These causal language models are a natural fit for generating text for a very practical, low-level reason: we can generate one word at a time from them, e.g., from left to right the way English writers do. This parallel to the human process of producing written language has lent a natural metaphor: that language models are trained to "write" in a manner akin to humans learning to write in grade school. This misplaced similarity has ended up obfuscating what models actually do: attempt to capture the distribution of possible continuations according to their likelihood as indicated by the training set.

Most models are trained with the Cross-Entropy loss $-\frac{1}{N}\sum_{i=0}^{N}\log p_\theta(x_i)$ (Murphy, 2012) which, when perfectly optimized with sufficient data, leads to a model that exactly captures actual distribution of the data.

There are, however, two major issues:

1. There is never sufficient data to properly determine the underlying distribution.
2. Model specific inductive biases warp the learned distribution in systematic ways.

Point 1 stems from the fact that language is an incredibly complex phenomenon that encodes layers of nested meaning (Kilgarriff, 2005), which makes it unlikely that there is enough data to lower the noise floor sufficiently to understand every aspect of language. Point 2 comes from the fact that specific models can learn certain patterns more easily, as demonstrated by the prevalence of repetition loops (Holtzman et al., 2020), misrepresentation of the ngram distribution (McCoy et al., 2023), and other effects that suggest that, while it is possible models may eventually converge to an approximation of the distribution given enough data, current models are consistently misrepresenting aspects of the training data. The basic tension is that training models more on the same data results in memorization (Tirumala et al., 2022), making it difficult to train models sufficiently to fully capture the distribution before memorization takes over. In practice, training sets and models are now so large that training for multiple epochs is often infeasible, and the relationship between memorization and over-fitting appears more complicated than has traditionally been assumed.

In Chapter 2, we address the fact that the bias language models have towards copying from text they are conditioned on and over-representing rare events cause strange and undesirable behavior that we call *neural text degeneration*. In addition to characterizing this issue, we show that contemporary methods to alleviate it such as Top-k sampling (Fan, Lewis, and Dauphin, 2018) have some success, but result in new problems. Therefore, we propose Nucleus (Top-p) Sampling to help close the gap and stop models from degenerating as easily.

1.3 ARE LANGUAGE MODELS SIMULATING QUESTION ANSWERERS?

Very soon after powerful causal and masked language models became generally available, such as GPT-2 (Radford et al., 2019a) and BERT (Devlin et al., 2019), it was discovered how useful these models were as queryable stores of knowledge (Petroni et al., 2019). While the factuality of the knowledge available from language models is far from from perfect, their coverage is often far superior to those of traditional knowledge bases due to the extensive amount of unstructured text available on the public internet.

This interpretation of models as factorized stores of human knowledge, echoes previous attempts in AI to structure all human knowledge, epitomized by Cyc (Lenat, 1995). However, given that language models assign all strings positive probability, and that generated answers are expensive to evaluate with human experts, evaluating "what language models know" has become a key issue (Jiang et al., 2020b).

One popular way to evaluate and use the implicit knowledge in such language models is to score multiple possible answers to a question, i.e., in multiple choice (or classification, where the question is understood to be "What class does this sample fit best in?"). Unfortunately, it has been shown that NLP benchmarks are riddled with artifacts, features that allow models to solve multiple choice problems without solving the underlying task that experimenters were attempting to test. Language models' inability to capture the intended task can be shown by removing what is considered to be necessary information (Cai, Tu, and Gimpel, 2017; Gururangan et al., 2018; Poliak et al., 2018, *inter alia*) or by inserting irrelevant information that causes a model to fail (Jia and Liang, 2017; Nie et al., 2020; Wallace et al., 2019, *inter alia*).

The reverse also happens: we *underestimate* model performance. While impressive performance has been achieved by interpreting the highest probability answer as the model's "chosen" answer (Brown et al., 2020), language models are fundamentally a distribution over possible strings, and the result is that language models capture *the distribution of possible answers* when used for QA, rather than a singular option. In other words, ranking by string probability can be problematic due to surface form competition-wherein different surface forms compete for probability mass, even if they represent the same underlying concept in the current context, e.g. "computer" and "PC." As probability mass is finite, this lowers the probability of the correct answer, due to competition from other valid answers outside of the multiple choice options. The result is that language models are much more likely to generate a correct answer, but that the probabilities of different answers are more calibrated to how *frequently* an answer would be given than to its correctness.

Chapter 3 studies this effect, which we name *surface form competition*, in which different surface forms for a concept (e.g. "US" and "the United States") compete for finite probability mass. This separates the probability of an answer to a question from its plausibility. We introduce Domain Conditional Pointwise Mutual Information, an alternative scoring function to directly compensate for surface form competition by reweighting each option according to a term that is proportional to its a priori likelihood within the given task. It achieves consistent gains in zero-shot performance over both calibrated (Zhao et al., 2021) and uncalibrated scoring functions.

1.4 ARE LANGUAGE MODELS SIMULATING COMMUNICATION?

Humans do not generate text arbitrarily. Rather text is communication, used to achieve pragmatic goals (Tomasello, 2005). Language models do not learn the same way as humans, and while it is natural to attempt to use them to construct language, this raises the question: are language models properly modeling human communicative goals?

Recent work has argued for viewing language models as agent models (Andreas, 2022), attempting to mimic an inferred author implied by preceding text. This appears to be broadly correct, with models capturing the intentions of authors more and more accurately with increased data, and better human preference modeling (Schulman et al., 2022; Ziegler et al., 2019) resulting in more language-model based systems deployed to users. Yet a gap still remains on many basic aspects underlying human communication that require consideration of *the reader's mental state* (Ullman, 2023).

This is, perhaps, not too surprising: language models are trained to capture the density of human language in the space of all possible text that could be written. As mentioned in §1.2, the given data almost always under-determines real human behavior, resulting in models that do not properly capture the goals of real humans. The question is then, how can we formalize these goals in a way where we can teach models to abide by them? Grice, Cole, Morgan, et al. (1975) note four objectives that humans usually optimize for when communicating:

1. *Quantity*: Be as informative as necessary, but no more.
2. *Quality*: Do not state what you believe is false or that which lacks evidence.
3. *Relevance*: Communicate information which is pertinent to the current exchange.
4. *Clarity*: Be clear, brief, and orderly, avoiding ambiguity and obscurity.

These by no means cover all of the aspects of human communication, they simply underpin assumptions present in most communicative contexts. While generative models pick up on surface-level features of these principles, they often do not enforce them directly, e.g., as showcased by "infelicitous" generated texts from models that are not grammatically wrong, so much as disconnected from a meaningful intent that makes the utterance useful to the reader.

Chapter 4 notes that, despite their local fluency, long-form text generated from language models is often generic, repetitive, and even self-contradictory. Within this chapter, we propose a unified learning framework that collectively addresses all the above issues by composing a committee of discriminators that can guide a base generator towards more globally coherent generations. More concretely, discriminators each specialize in a different principle of communication, such as Grice's maxims, and are collectively combined with the base language model through a composite decoding objective. Human evaluation demonstrates that text generated by our model is preferred over that of baselines by a large margin, significantly enhancing the overall coherence, style, and information of the generations.

1.5 WHAT, PRECISELY, ARE LANGUAGE MODELS DOING?

One of the most common interpretative mistakes when approaching computational models is that they do what we wanted them to do, poorly. More common is that one's specification for the model is incorrect and the resulting model is simply doing something else entirely. Coaxing out desired behavior from pretrained models, while avoiding undesirable ones has redefined NLP and is reshaping how we interact with computers. What was once a scientific engineering discipline—in which building blocks are stacked one on top of the other—is arguably already a complex systems science—in which *emergent behaviors* are sought out to support previously unimagined use cases.

Despite the ever increasing number of benchmarks that measure *task performance*, we lack explanations of what *behaviors* language models exhibit that allow them to complete these tasks in the first place.

This is a challenge, because it requires us to fundamentally change our attitude towards the kind of science we do. Traditionally, machine learning is largely about understanding how different models, trained and used in various ways, can generalize to a training distribution or even a related distribution that they were not directly trained on. The task of understanding *an already trained model* is much different, a problem more akin to biology: as if a totally new species was discovered with some strange new property, like a complex set of sounds it emits in different contexts, that researchers must discover the mechanics of.

Previously, when newly proposed machine learning methods have produced models we do not understand, we have used *ablations* to isolate the specific point of change between models we do understand and models we don't understand. Once the change that made the difference has been isolated, many parallel models trained with slightly different details could be trained to probe the effect on models and characterize the effect of new methods. However, this is no longer practical with language models, which have become overwhelmingly expensive to train, costing half a million dollars to train a state-of-the-art model from three years ago (Venigalla and Li, 2022).

Instead, models must be examined more as artifacts that cannot be reproduced, the way planets are studied in astronomy. This, of course, does not mean we cannot run experiments—but that these experiments are meant to analyze what models are already doing, rather than what a different model trained a different way would do. This is often how we approach the classic media generator, human beings, which cannot be retrained for the whims of an experimenter. Yet, despite the inability to retrain massive language models for study by ablation, studying a computational model directly rather than an organic object allows us to experiment more easily than with most complex systems, which are often difficult to record and perturb.

Chapter 5 addresses these issues, pointing out our insufficient vocabulary for describing the more basic elements of non-formal tasks as simple as email rewriting and proposing a systematic effort to decompose language model behavior into categories that explain cross-task performance, to guide mechanistic explanations and help future-proof analytic research.

1.6 SCOPE OF THIS DISSERTATION

The preceding sections formed an overview of the aspects of generative models of text that this dissertation will address, focusing in large part on revealing what we don't know about the current workhorse of NLP: language models. On the practical side, this lack of understanding is dangerous: as language model systems are increasingly deployed (Eloundou et al., 2023; George and George, 2023; Ray, 2023), our inability to predict, correct, or even properly notice the error modes of these models present an increasing risk to society (Bender et al., 2021; Bommasani et al., 2021). We would also like to present another perspective, that while we should be increasingly careful about the deployment of models that we often think are more controllable and auditable than they are (Turpin et al., 2023), we should equally bring our scientific curiosity to the table and ask: what explains model behavior? What working model should we have in our head for how they operate? What heuristics can we "catch" models relying on?

This dissertation attempts to channel this scientific mindset to understand how we should interpret language models and, looking towards the future, generative models of human media.l

The chapters are as follows:

CORRECTING CORRECTNESS

- Chapter 2 presents **Neural Text Degeneration**, a study of the characteristics, cause, and solution for certain decoding errors in language models.

  > This chapter was previously published as: Ari Holtzman, Jan Buys, Maxwell Forbes, and Yejin Choi (2020). "The Curious Case of Neural Text Degeneration." In: *International Conference on Learning Representations*.

- Chapter 3 presents **Surface Form Competition**, a project that identifies a significant source of error in a standard method for question answering with language models, then proposes an underlying theory and solution.

  > This chapter was previously published as: Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer (2021). "Surface Form Competition: Why the Highest Probability Answer Isn't Always Right." In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7038–7051.

MODELING MODELS

- Chapter 4 presents **Learning to Write**, in which Grice's Maxims are encoded as collaborative models for improving text generation through a search procedure meant to model human communication more closely.

> This chapter was previously published as: Ari Holtzman, Jan Buys, Maxwell Forbes, Antoine Bosselut, David Golub, and Yejin Choi (2018). "Learning to Write with Cooperative Discriminators." In: *Proceedings of the Association for Computational Linguistics*.

- Chapter 5 presents **Generative Models as a Complex Systems Science**, which proposes a new framework for studying generative models moving forward.

> At the time of writing, this chapter is not published, but a preprint has been published as: Ari Holtzman, Peter West, and Luke Zettlemoyer (2023). "Generative Models as a Complex Systems Science: How can we make sense of large language model behavior?" In: *preprint*.

Part I

CORRECTING CORRECTNESS

2

NEURAL TEXT DEGENERATION

This section is adapted from Holtzman et al. (2020)

Context: In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

Beam Search, *b*=32:
"The study, published in the Proceedings of the National Academy of Sciences of the United States of America (PNAS), was conducted by researchers from the Universidad Nacional Autónoma de México (UNAM) and the Universidad Nacional Autónoma de México (UNAM/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de ..."

Pure Sampling:
They were cattle called Bolivian Cavalleros; they live in a remote desert uninterrupted by town, and they speak huge, beautiful, paradisiacal Bolivian linguistic thing. They say, 'Lunch, marge.' They don't tell what the lunch is," director Professor Chuperas Omwell told Sky News. "They've only been talking to scientists, like we're being interviewed by TV reporters. We don't even stick around to be interviewed by TV reporters. Maybe that's how they figured out that they're cosplaying as the Bolivian Cavalleros."

Figure 2.1: Even with substantial human context and the powerful GPT-2 Large language model, Beam Search (size 32) leads to degenerate repetition (highlighted in blue) while pure sampling leads to incoherent gibberish (highlighted in red). When $b \geq 64$, both GPT-2 Large and XL (774M and 1542M parameters, respectively) prefer to stop generating immediately after the given context.

On February 14th 2019, OpenAI surprised the scientific community with an impressively high-quality article about Ovid's Unicorn, written by GPT-2.[1] Notably, the top-quality generations obtained from the model rely on *randomness* in the decoding method, in particular through *top-k* sampling that samples the next word from the top k most probable choices (Fan, Lewis, and Dauphin, 2018; Holtzman et al., 2018; Radford et al., 2019a), instead of aiming to decode text that *maximizes* likelihood.

In fact, decoding strategies that optimize for output with high probability, such as beam search, lead to text that is incredibly degenerate, even when using state-of-the-art models such as GPT-2 Large, as shown in Figure 2.1. This may seem counter-intuitive, as one would expect that good models would assign higher probability to more human-like, grammatical text. Indeed, language models do generally assign high scores to well-formed text, yet the *highest* scores for longer texts are often generic, repetitive, and awkward. Figure 2.2 exposes how different the distribution of probabilities assigned to beam search decoded text and naturally occurring text really are. This is an *interpretation error*: likelihood was assumed to imply coherence, but is measuring something quite different.

Perhaps equally surprising is the right side of Figure 2.1, which shows that pure sampling—sampling directly from the probabilities predicted by the model—results in text that is incoherent and almost unrelated to the context. Why is text produced by pure sampling so degenerate?

In this work we show that the "unreliable tail" is to blame. This unreliable tail is composed of tens of thousands of candidate tokens with relatively low probability that are over-represented in the aggregate.

1 https://openai.com/blog/better-language-models/

To overcome these issues we introduce *Nucleus Sampling* (§2.2.1). The key intuition of Nucleus Sampling is that the vast majority of probability mass at each time step is concentrated in the *nucleus*, a small subset of the vocabulary that tends to range between one and a thousand candidates. Instead of relying on a fixed top-k, or using a temperature parameter to control the shape of the distribution without sufficiently suppressing the unreliable tail, we propose sampling from the top-p portion of the probability mass, expanding and contracting the candidate pool dynamically.

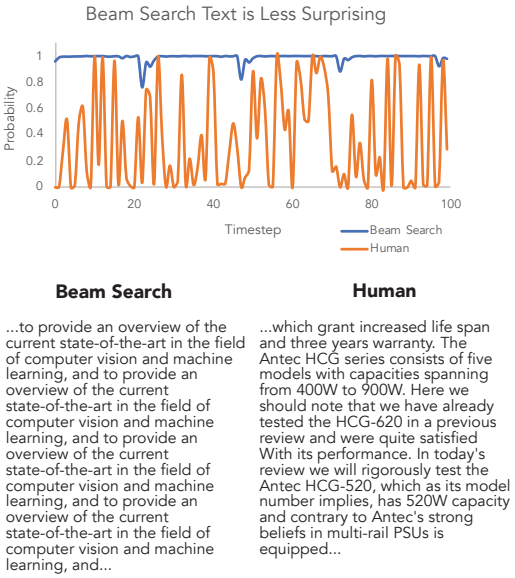

Figure 2.2: The probability assigned to tokens generated by Beam Search and humans, given the same context. Note the increased variance that characterizes human text, in contrast with the endless repetition of text decoded by Beam Search.

In order to compare current methods to Nucleus Sampling, we compare various distributional properties of generated text to the reference distribution, such as the likelihood of veering into repetition and the perplexity of *generated* text. The latter reveals that text generated by maximization or top-k sampling is *too* probable, indicating a lack of diversity and divergence in vocabulary usage from the human distribution. On the other hand, pure sampling produces text that is significantly *less* likely than the gold, corresponding to lower generation quality.

Vocabulary usage and Self-BLEU (Zhu et al., 2018) statistics reveal that high values of k are needed to make top-k sampling match human statistics. Yet, generations based on high values of k often have high variance in likelihood, hinting at qualitatively observable incoherency issues. Nucleus Sampling can easily match reference perplexity through tuning the value of p, avoiding the incoherence caused by setting k high enough to match distributional statistics.

Finally, we perform Human Unified with Statistical Evaluation (HUSE; Hashimoto, Zhang, and Liang, 2019) to jointly assess the overall quality and diversity of the decoding strategies, which cannot be captured using either human or automatic evaluation alone. The HUSE evaluation demonstrates that Nucleus Sampling is the best overall decoding strategy. We include generated examples for qualitative analysis—see Figure 2.3 for a representative example.

2.1 BACKGROUND

2.1.1 *Text Generation Decoding Strategies*

A number of recent works have alluded to the disadvantages of generation by maximization, which tend to generate output with high grammaticality but low diversity (Fan, Lewis, and Dauphin, 2018; Holtzman et al., 2018; Kulikov et al., 2018). Generative Adversarial Networks (GANs) have been a prominent research direction (Xu et al., 2018; Yu et al., 2017a), but recent work has shown that when quality and diversity are considered jointly, GAN-generated text fails to outperform generations from language models (Caccia et al., 2018; Semeniuta, Severyn, and Gelly, 2018; Tevet et al., 2018). Work on neural dialog systems have proposed methods for diverse beam search, using a task-specific diversity scoring function or constraining beam hypotheses to be sufficiently different (Kulikov et al., 2018; Li, Monroe, and Jurafsky, 2016; Vijayakumar et al., 2018). While such utility functions encourage desirable properties in generations, they do not remove the need to choose an appropriate decoding strategy, and we believe that Nucleus Sampling will have complementary advantages in such approaches. Finally, (Welleck et al., 2019) begin to address the problem of neural text degeneration through an "unlikelihood loss", which decreases training loss on repeated tokens and thus implicitly reduces gradients on frequent tokens as well. Our focus is on exposing neural text degeneration and providing a *decoding* solution that can be used with arbitrary models, but future work will likely combine training-time and inference-time solutions.

2.1.2 *Open-ended vs directed generation*

Many text generation tasks are defined through (input, output) pairs, such that the output is a constrained *transformation* of the input. Example applications include machine translation (Bahdanau, Cho, and Bengio, 2015a), data-to-text generation (Wiseman, Shieber, and Rush, 2017), and summarization (Nallapati et al., 2016). We refer to these tasks as *conditional* generation. Typically encoder-decoder architectures are used, often with an attention mechanism (Bahdanau, Cho, and Bengio, 2015a; Luong, Pham, and Manning, 2015) or using attention-based architectures such as the Transformer (Vaswani et al., 2017). Generation is usually performed using beam search; since output is tightly scoped by the input, repetition and genericness are not as problematic. Still, similar issues have been reported when using large beam sizes (Koehn and Knowles, 2017) and more recently with exact inference (Stahlberg and Byrne, 2019a), a counter-intuitive observation since more comprehensive search helps maximize probability.

Open-ended generation, which includes conditional story generation and contextual text continuation (as in Figure 2.1), has recently become a promising research direction due to significant advances in neural language models (Clark, Ji, and Smith, 2018; Fan, Lewis, and Dauphin, 2018; Holtzman et al., 2018; Peng et al., 2018; Radford et al., 2019a). While the input context restricts the space of acceptable output generations, there is a considerable degree of freedom in what can plausibly come next, unlike in directed

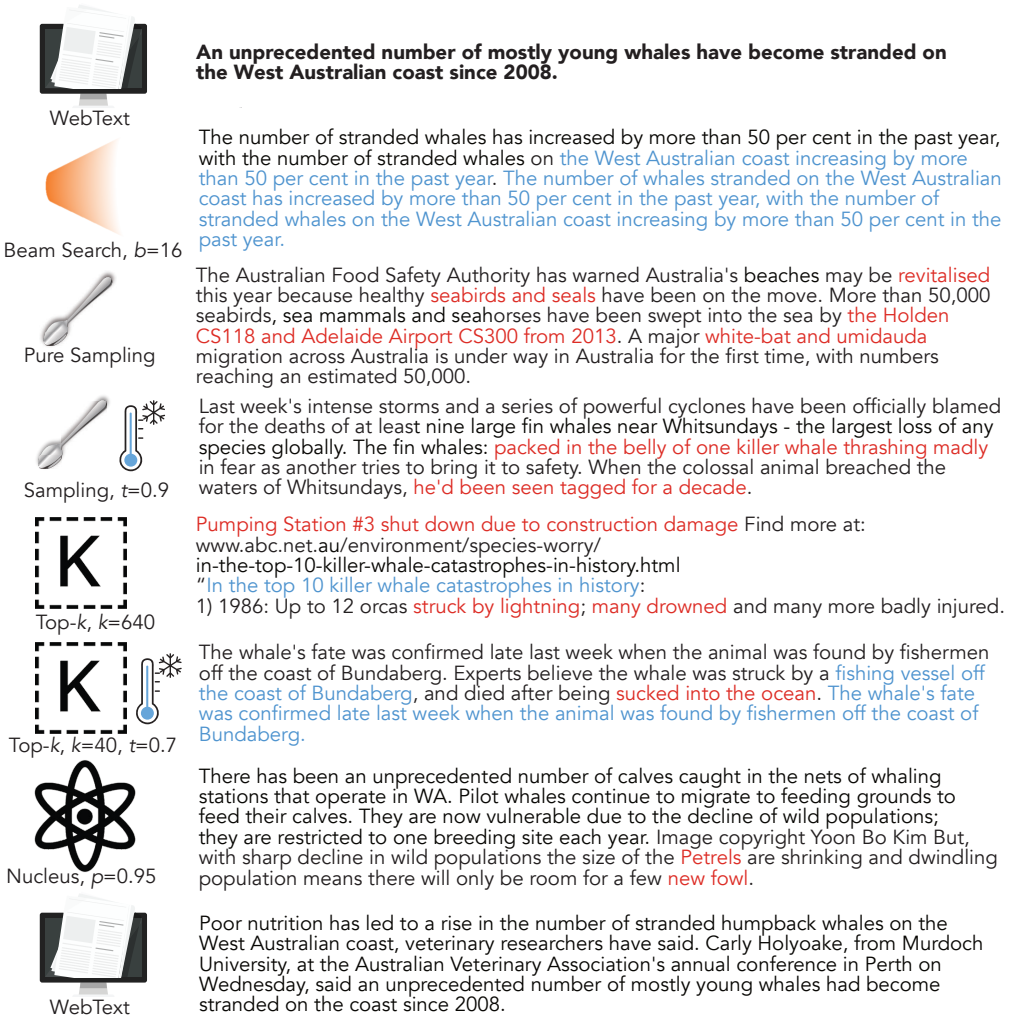

Figure 2.3: Example generations continuing an initial sentence. Maximization and top-k truncation methods lead to copious repetition (highlighted in blue), while sampling with and without temperature tends to lead to incoherence (highlighted in red). Nucleus Sampling largely avoids both issues.

generation settings. Our work addresses the challenges faced by neural text generation with this increased level of freedom, but we note that some tasks, such as goal-oriented dialog, may fall somewhere in between open-ended and directed generation.

2.2 LANGUAGE MODEL DECODING

Given an input text passage as context, the task of *open-ended* generation is to generate text that forms a coherent continuation from the given context. More formally, given a sequence of m tokens $x_1 \ldots x_m$ as **context**, the task is to generate the next n **continuation** tokens to obtain the completed sequence $x_1 \ldots x_{m+n}$. We assume that

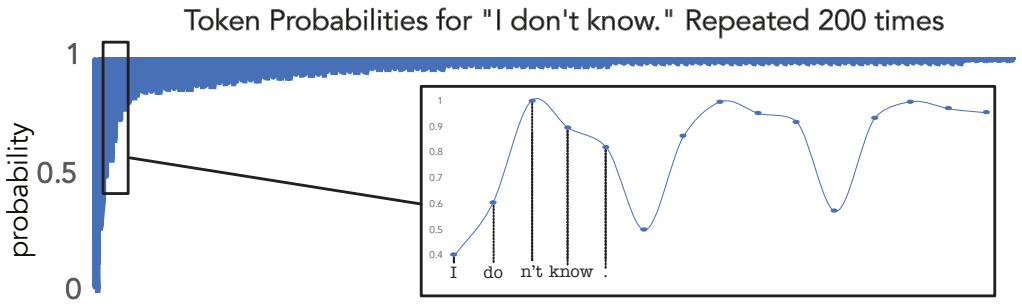

Figure 2.4: The probability of a repeated phrase increases with each repetition, creating a positive feedback loop. We found this effect to hold for the vast majority of phrases we tested, regardless of phrase length or if the phrases were sampled randomly rather than taken from human text.

models compute $P(x_{1:m+n})$ using the common left-to-right decomposition of the text probability,

$$P(x_{1:m+n}) = \prod_{i=1}^{m+n} P(x_i|x_1 \dots x_{i-1}), \tag{2.1}$$

which is used to generate the generation token-by-token using a particular *decoding strategy*.

MAXIMIZATION-BASED DECODING The most commonly used decoding objective, in particular for conditional generation, is maximization-based decoding. Assuming that the model assigns higher probability to higher quality text, these decoding strategies search for the continuation with the highest likelihood. Since finding the optimum argmax sequence from recurrent neural language models or Transformers is not tractable (Chen et al., 2018), common practice is to use beam search (Li et al., 2016b; Shen et al., 2017; Wiseman, Shieber, and Rush, 2017). However, several recent studies on open-ended generation have reported that maximization-based decoding does not lead to high quality text (Fan, Lewis, and Dauphin, 2018; Holtzman et al., 2018).

2.2.1 *Nucleus Sampling*

We propose a new stochastic decoding method: Nucleus Sampling. The key idea is to use the shape of the probability distribution to determine the set of tokens to be sampled from. Given a distribution $P(x|x_{1:i-1})$, we define its top-p vocabulary $V^{(p)} \subset V$ as the smallest set such that

$$\sum_{x \in V^{(p)}} P(x|x_{1:i-1}) \geq p. \tag{2.2}$$

Let $p' = \sum_{x \in V^{(p)}} P(x|x_{1:i-1})$. The original distribution is re-scaled to a new distribution, from which the next word is sampled:

$$P'(x|x_{1:i-1}) = \begin{cases} P(x|x_{1:i-1})/p' & \text{if } x \in V^{(p)} \\ 0 & \text{otherwise.} \end{cases} \tag{2.3}$$

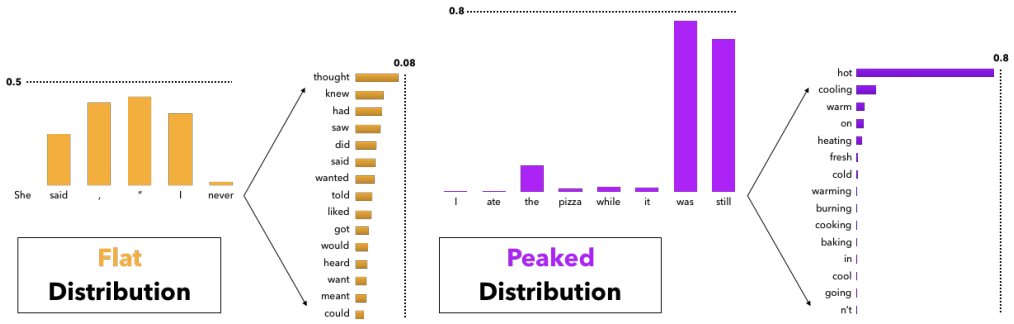

Figure 2.5: The probability mass assigned to partial human sentences. Flat distributions lead to many moderately probable tokens, while peaked distributions concentrate most probability mass into just a few tokens. The presence of flat distributions makes the use of a small k in top-k sampling problematic, while the presence of peaked distributions makes large k's problematic.

In practice this means selecting the highest probability tokens whose cumulative probability mass exceeds the pre-chosen threshold p. The size of the sampling set will adjust dynamically based on the shape of the probability distribution at each time step. For high values of p, this is a small subset of vocabulary that takes up vast majority of the probability mass — the *nucleus*.

2.2.2 *Top-k Sampling*

Top-k sampling has recently become a popular alternative sampling procedure (Fan, Lewis, and Dauphin, 2018; Holtzman et al., 2018; Radford et al., 2019a). Nucleus Sampling and top-k both sample from truncated Neural LM distributions, differing only in the strategy of where to truncate. Choosing where to truncate can be interpreted as determining the generative model's trustworthy prediction zone.

At each time step, the top k possible next tokens are sampled from according to their relative probabilities. Formally, given a distribution $P(x|x_{1:i-1})$, we define its top-k vocabulary $V^{(k)} \subset V$ as the set of size k which maximizes $\sum_{x \in V^{(k)}} P(x|x_{1:i-1})$. Let $p' = \sum_{x \in V^{(k)}} P(x|x_{1:i-1})$. The distribution is then re-scaled as in equation 2.3, and sampling is performed based on that distribution. Note that the scaling factor p' can vary wildly at each time-step, in contrast to Nucleus Sampling.

DIFFICULTY IN CHOOSING A SUITABLE VALUE OF k While top-k sampling leads to considerably higher quality text than either beam search or sampling from the full distribution, the use of a constant k is sub-optimal across varying contexts. As illustrated on the left of Figure 2.5, in some contexts the head of the next word distribution can be flat across tens or hundreds of reasonable options (e.g. nouns or verbs in generic contexts), while in other contexts most of the probability mass is concentrated in one or a small number of tokens, as on the right of the figure. Therefore if k is small, in some contexts there is a risk of generating bland or generic text, while if k is large the top-k vocabulary will include inappropriate candidates which will have their probability of being sampled *increased* by the renormalization. Under Nucleus Sampling, the number

of candidates considered rises and falls dynamically, corresponding to the changes in the model's confidence region over the vocabulary which top-k sampling fails to capture for any one choice of k.

2.2.3 *Sampling with Temperature*

Another common approach to sampling-based generation is to shape a probability distribution through temperature (Ackley, Hinton, and Sejnowski, 1985). Temperature sampling has been applied widely to text generation (Caccia et al., 2018; Fan, Lewis, and Dauphin, 2018; Ficler and Goldberg, 2017). Given the logits $u_{1:|V|}$ and temperature t, the softmax is re-estimated as

$$p(x = V_l | x_{1:i-1}) = \frac{\exp(u_l/t)}{\sum_{l'} \exp(u'_l/t)}. \tag{2.4}$$

Setting $t \in [0, 1)$ skews the distribution towards high probability events, which implicitly lowers the mass in the tail distribution. Low temperature sampling has also been used to partially alleviate the issues of top-k sampling discussed above, by shaping the distribution before top-k sampling (Fan, Lewis, and Dauphin, 2018; Radford et al., 2018b). However, recent analysis has shown that, while lowering the temperature improves generation quality, it comes at the cost of decreasing diversity (Caccia et al., 2018; Hashimoto, Zhang, and Liang, 2019).

2.3 LIKELIHOOD EVALUATION

2.3.1 *Experimental Setup*

While many neural network architectures have been proposed for language modeling, including LSTMs (Sundermeyer, Schlüter, and Ney, 2012) and convolutional networks (Dauphin et al., 2017), the Transformer architecture (Vaswani et al., 2017) has been the most successful in the extremely large-scale training setups in recent literature (Radford et al., 2018b; 2019a). In this study we use the Generatively Pre-trained Transformer, version 2 (GPT2; a), which was trained on WebText, a 40GB collection of text scraped from the web.[2] We perform experiments using the Large model (762M parameters). Our analysis is based on generating 5,000 text passages, which end upon reaching an end-of-document token or a maximum length of 200 tokens. Texts are generated conditionally, conditioned on the initial paragraph (restricted to 1-40 tokens) of documents in the held-out portion of WebText, except where otherwise mentioned.

2.3.2 *Perplexity*

Our first evaluation is to compute the perplexity of *generated* text using various decoding strategies, according to the model that is being generated from. We compare these

2 Available at https://github.com/openai/gpt-2-output-dataset

Method	Perplexity	Self-BLEU4	Zipf Coefficient	Repetition %	HUSE
Human	12.38	0.31	0.93	0.28	-
Greedy	1.50	0.50	1.00	73.66	-
Beam, b=16	1.48	0.44	0.94	28.94	-
Stochastic Beam, b=16	19.20	0.28	0.91	0.32	-
Pure Sampling	22.73	0.28	**0.93**	0.22	0.67
Sampling, t=0.9	10.25	0.35	0.96	0.66	0.79
Top-k=40	6.88	0.39	0.96	0.78	0.19
Top-k=640	13.82	**0.32**	0.96	**0.28**	0.94
Top-k=40, t=0.7	3.48	0.44	1.00	8.86	0.08
Nucleus p=0.95	**13.13**	**0.32**	0.95	0.36	**0.97**

Table 2.1: Main results for comparing all decoding methods with selected parameters of each method. The numbers *closest to human scores* are in **bold** except for HUSE (), a combined human and statistical evaluation, where the highest (best) value is **bolded**. For Top-k and Nucleus Sampling, HUSE is computed with interpolation rather than truncation (see §2.5.1).

perplexities against that of the gold text (Figure 2.6). Importantly, we argue that the optimal generation strategy should produce text which has a perplexity *close to* that of the gold text: Even though the model has the ability to generate text that has lower perplexity (higher probability), such text tends to have low diversity and get stuck in repetition loops, as shown in §2.4 and illustrated in Figure 2.4.

We see that perplexity of text obtained from pure sampling is *worse* than the perplexity of the gold. This indicates that the model is confusing itself: sampling too many unlikely tokens and creating context that makes it difficult to recover the human distribution of text, as in Figure 2.1. Yet, setting the temperature lower creates diversity and repetition issues, as we shall see in §2.4. Even with our relatively fine-grained parameter sweep, Nucleus Sampling obtains closest perplexity to human text, as shown in Table 2.1.

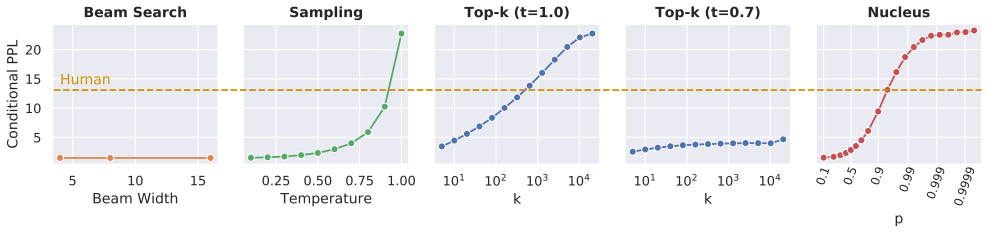

Figure 2.6: Perplexities of generations from various decoding methods. Note that beam search has unnaturally low perplexities. A similar effect is seen using a temperature of 0.7 with top-k as in both (Radford et al., 2019a) and (Fan, Lewis, and Dauphin, 2018). Sampling, Top-k, and Nucleus can all be calibrated to human perplexities, but the first two face coherency issues when their parameters are set this high.

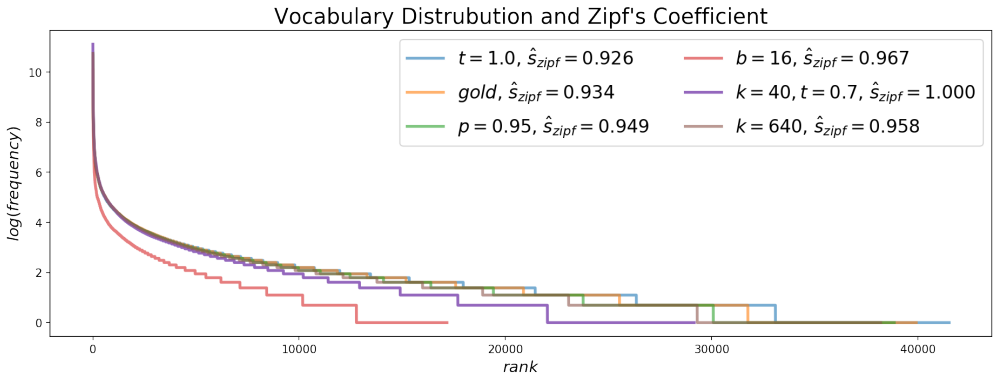

Figure 2.7: A rank-frequency plot of the distributional differences between n-gram frequencies of human and machine text. Sampling and Nucleus Sampling are by far the closest to the human distribution, while Beam Search clearly follows a very different distribution than natural language.

2.3.3 *Natural Language Does Not Maximize Probability*

One might wonder if the issue with maximization is a *search error*, i.e., there are higher quality sentences to which the model assigns higher probability than to the decoded ones, beam search has just failed to find them. Yet Figures 2.2 & 2.6 show that the per-token probability of natural text is, on average, much *lower* than text generated by beam search. Natural language rarely remains in a high probability zone for multiple consecutive time steps, instead veering into lower-probability but more informative tokens. Nor does natural language tend to fall into repetition loops, even though the model tends to assign high probability to this, as seen in Figure 2.4.

We contend that the failures of maximization based decoding are the result of an *interpretation error*: a misunderstanding about what high likelihood signifies.

Why is human-written text *not* the most probable text? We conjecture that this is an intrinsic property of human language. Language models that assign probabilities one word at a time without a global model of the text will have trouble capturing this effect. Grice's Maxims of Communication (Grice, 1975) show that people optimize against stating the obvious. Thus, making every word as predictable as possible will be disfavored. This makes solving the problem simply by training larger models or improving neural architectures using standard per-word learning objectives unlikely: such models are forced to favor the lowest common denominator, rather than informative language.

2.4 DISTRIBUTIONAL STATISTICAL EVALUATION

2.4.1 *Zipf Distribution Analysis*

In order to compare generations to the reference text, we begin by analyzing their use of vocabulary. Zipf's law suggests that there is an exponential relationship between the rank of a word and its frequency in text. The Zipfian coefficient s can be used to compare the distribution in a given text to a theoretically perfect exponential curve, where $s = 1$

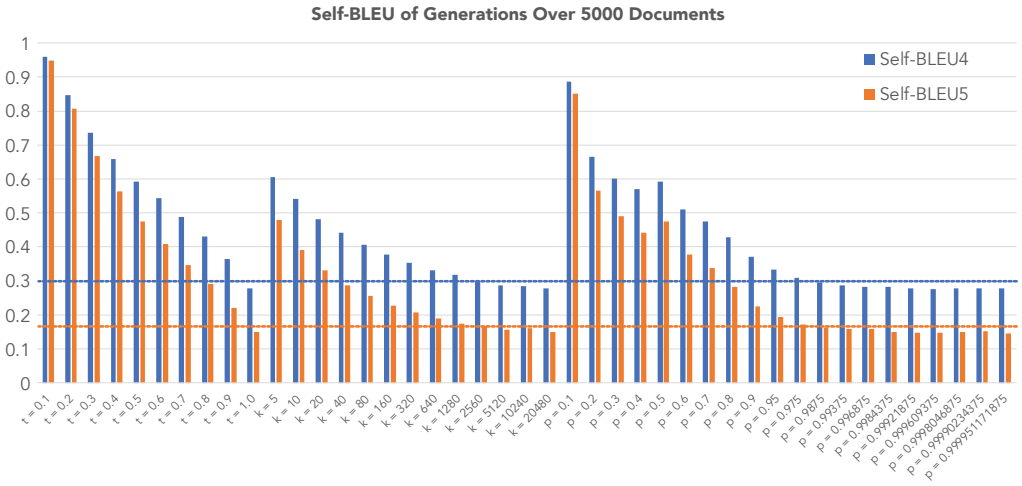

Figure 2.8: Self-BLEU calculated on the unconditional generations produced by stochastic decoding methods; lower Self-BLEU scores imply higher diversity. Horizontal blue and orange lines represent human self-BLEU scores. Note how common values of $t \in [0.5, 1]$ and $k \in [1, 100]$ result in high self-similarity, whereas "normal" values of $p \in [0.9, 1)$ closely match the human distribution of text.

(Piantadosi, 2014). Figure 2.7 shows the vocabulary distributions along with estimated Zipf coefficients for selected parameters of different decoding methods. As expected, pure sampling is the closest to the human distribution, followed by Nucleus Sampling. The visualization of the distribution shows that pure sampling slightly *overestimates* the use of rare words, likely one reason why pure sampling also has higher perplexity than human text. Furthermore, lower temperature sampling avoids sampling these rare words from the tail, which is why it has been used in some recent work (Fan, Lewis, and Dauphin, 2018; Radford et al., 2019a).

2.4.2 *Self-BLEU*

We follow previous work and compute Self-BLEU (Zhu et al., 2018) as a metric of diversity. Self-BLEU is calculated by computing the BLEU score of each generated document using *all other generations* in the evaluation set as references. Due to the expense of computing such an operation, we sample 1000 generations, each of which is compared with *all 4999 other generations as references.* A lower Self-BLEU score implies higher diversity. Figure 2.8 shows that Self-BLEU results largely follow that of the Zipfian distribution analysis as a diversity measure. It is worth noting that very high values of k and t are needed to get close to the reference distribution, though these result in unnaturally high perplexity (§2.3).

2.4.3 *Repetition*

One attribute of text quality that we can quantify is repetition. Figure 2.9 shows that Nucleus Sampling and top-k sampling have the least repetition for reasonable parameter

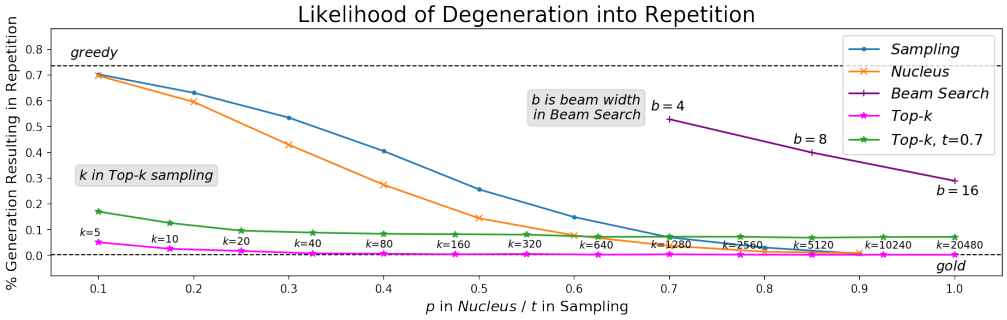

Figure 2.9: We visualize how often different decoding methods get "stuck" in loops within the first 200 tokens. A phrase (minimum length 2) is considered a repetition when it repeats at least **three** times at the *end* of the generation. We label points with their parameter values except for t and p which follow the x-axis. Values of k greater than 100 are rarely used in practice and values of p are usually in $[0.9, 1)$; therefore Nucleus Sampling is far closer to the human distribution in its usual parameter range. Sampling with temperatures lower than 0.9 severely increase repetition. Finally, although beam search becomes less repetitive according to this metric as beam width increases, this is largely because average length gets shorter as b increases.

ranges. Generations from temperature sampling have more repetition unless very high temperatures are used, which we have shown negatively affects coherence (as measured by high perplexity). Further, all stochastic methods face repetition issues when their tuning parameters are set too low, which tends to *over-truncate*, mimicking greedy search. Therefore we conclude that only Nucleus Sampling satisfies all the distributional criteria for desirable generations.

2.5 HUMAN EVALUATION

2.5.1 *Human Unified with Statistical Evaluation (HUSE)*

Statistical evaluations are unable to measure the coherence of generated text properly. While the metrics in previous sections gave us vital insights into the different decoding methods we compare, human evaluation is still required to get a full measure of the quality of the generated text. However, pure human evaluation does not take into account the diversity of the generated text; therefore we use HUSE (Hashimoto, Zhang, and Liang, 2019) to combine human and statistical evaluation. HUSE is computed by training a discriminator to distinguish between text drawn from the human and model distributions, based on only two features: The probability assigned by the language model, and human judgements of typicality of generations. Text that is close to the human distribution in terms of quality and diversity should perform well on both likelihood evaluation and human judgements.

As explored in the previous sections, the current best-performing decoding methods rely on *truncation* of the probability distribution, which yields a probability of 0 for the vast majority of potential tokens. Initial exploration of applying HUSE directly led to top-k and Nucleus Sampling receiving scores of nearly 0 due to truncation, despite

humans favoring these methods. As a proxy, when generating the text used to compute HUSE, we interpolate (with mass 0.1) the original probability distribution with the top-k and Nucleus Sampling distribution, smoothing the truncated distribution.

For each decoding algorithm we annotate 200 generations for typicality, with each generation receiving 20 annotations from 20 different annotators. This results in a total of 4000 annotations per a decoding scheme. We use a KNN classifier to compute HUSE, as in the original paper, with $k = 13$ neighbors, which we found led to the higher accuracy in discrimination. The results in Table 2.1 shows that Nucleus Sampling obtains the highest HUSE score, with Top-k sampling performing second best.

2.5.2 *Qualitative Analysis*

Figure 2.3 shows representative example generations. Unsurprisingly, beam search gets stuck in a repetition loop it cannot escape. Of the stochastic decoding schemes, the output of full sampling is clearly the hardest to understand, even inventing a new word "umidauda", apparently a species of bird. The generation produced by Nucleus Sampling isn't perfect – the model appears to confuse whales with birds, and begins writing about those instead. Yet, top-k sampling immediately veers off into an unrelated event. When top-k sampling is combined with a temperature of 0.7, as is commonly done (Fan, Lewis, and Dauphin, 2018; Radford et al., 2019a), the output devolves into repetition, exhibiting the classic issues of low-temperature decoding.

2.6 SUPPLEMENTARY A

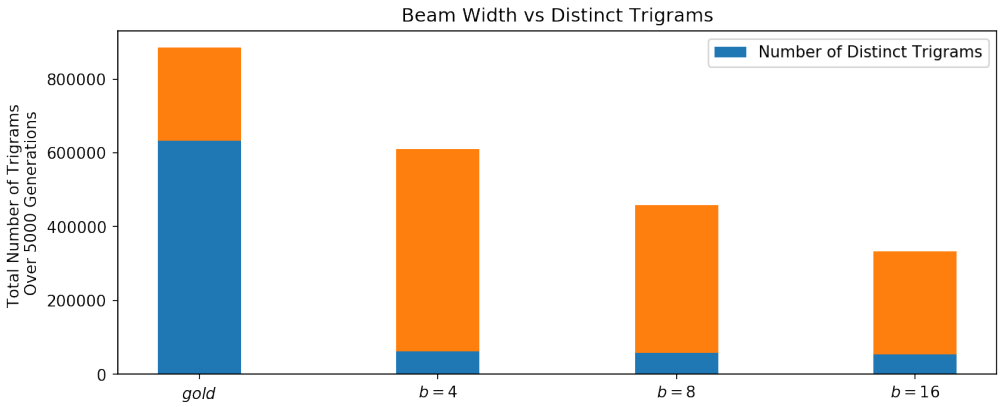

Figure 2.10: The total number of trigrams produced by Beam Search with varying beam widths, with gold (human) data for comparison. Note how the average length of generations goes down linearly with beam width, while the number of distinct trigrams stays constant and extremely low in comparison to gold data.

2.7 SUPPLEMENTARY B

We include a set of examples for further qualitative comparison.

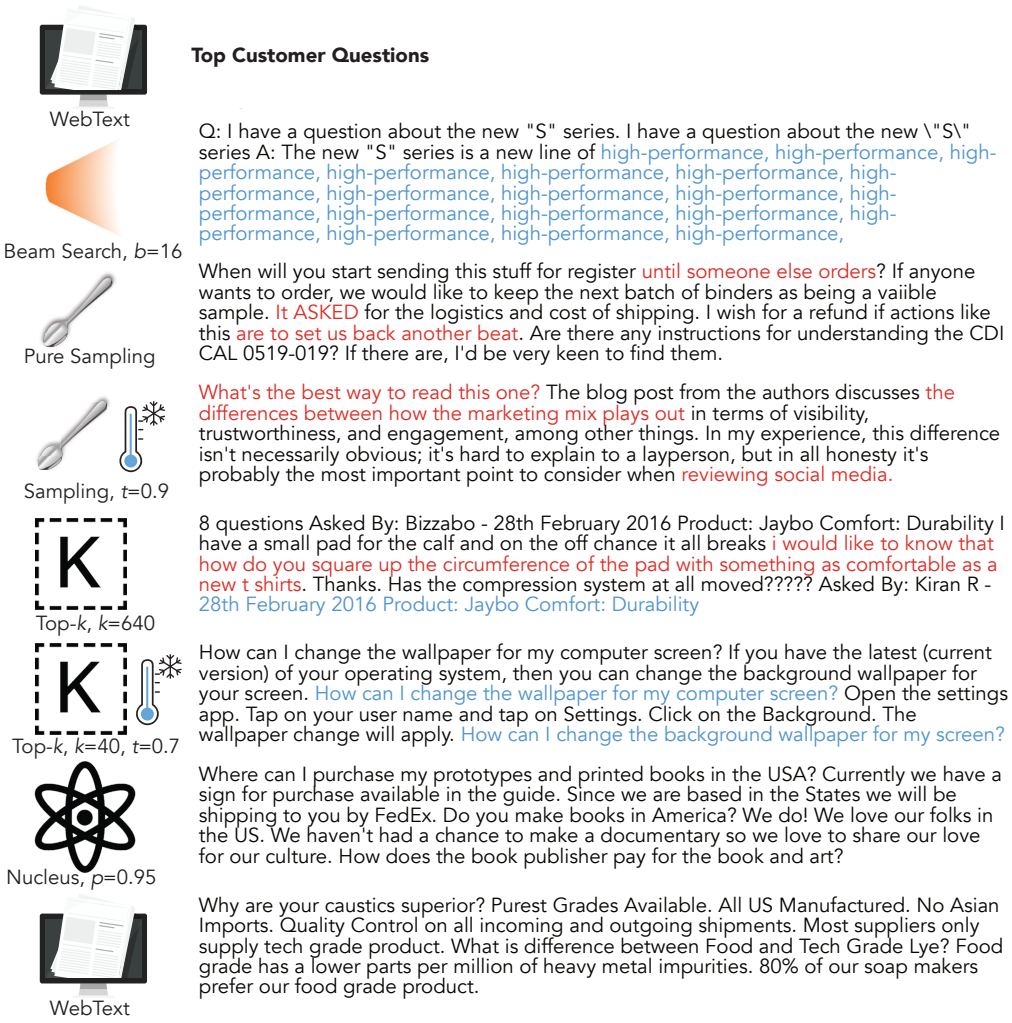

Figure 2.11: More example generations from an initial tag line.

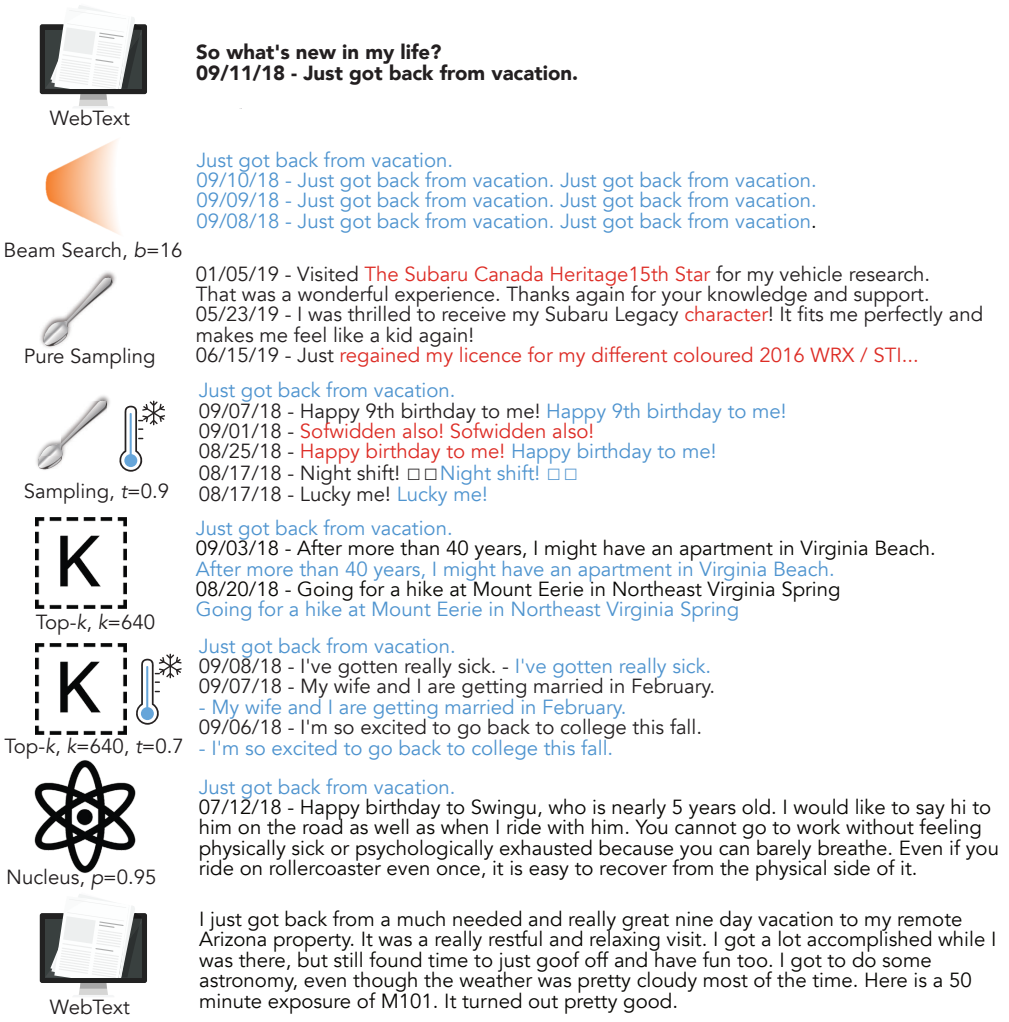

Figure 2.12: More example generations from an initial tag line. Note that Pure Sampling and Nucleus Sampling is the only algorithms that can escape the repetition loop, with Nucleus Sampling's generation far closer in style to the ground truth text.

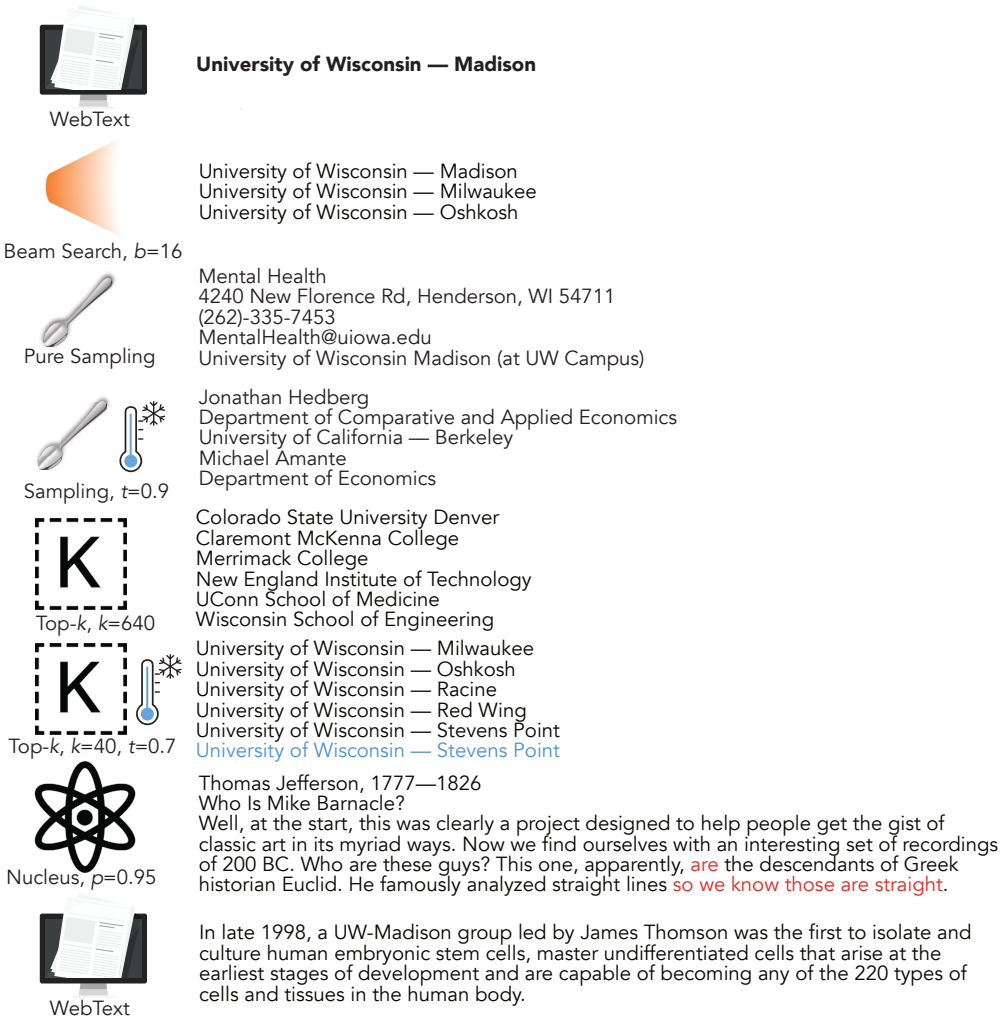

Figure 2.13: More example generations from an initial tag line.

3

SURFACE FORM COMPETITION

This section is adapted from Holtzman et al. (2021)

Despite the impressive results large pretrained language models have achieved in zero-shot settings (Brown et al., 2020; Radford et al., 2019c), we argue that current work underestimates the zero-shot capabilities of these models on classification tasks. This is in large part due to **surface form competition**—a property of generative models that causes probability to be rationed between different valid strings, even ones that differ trivially, e.g., by capitalization alone. Such competition can be largely removed by scoring choices according to Domain Conditional Pointwise Mutual Information (PMI_{DC}), which reweighs scores by how much *more* likely a hypothesis (answer) becomes given a premise (question) within the specific task domain.

Specifically, consider the example question (shown in Figure 3.1): "A human wants to submerge himself in water, what should he use?" with multiple choice options "Coffee cup", "Whirlpool bath", "Cup", and "Puddle." From the given options, "Whirlpool bath" is the only one that makes sense. Yet, other answers are valid and easier for a language model to generate, e.g., "Bathtub" and "A bathtub." Since all surface forms compete for finite probability mass, allocating significant probability mass to "Bathtub" decreases the amount of probability mass assigned to "Whirlpool bath." While the total probability of generating *some correct answer* may be high (i.e., across all valid surface forms), only one of these is a listed option. This is particularly problematic here, because "Whirlpool bath" will be much lower probability than "Bathtub," due to its rarity. More generally, methods that do not account for surface form competition will favor answers with fewer lexical paraphrases. This is an example of an *interpretation error*—probability assigned by a generative model was assumed to be a surrogate for "correctness" but has properties that don't match this interpretation.

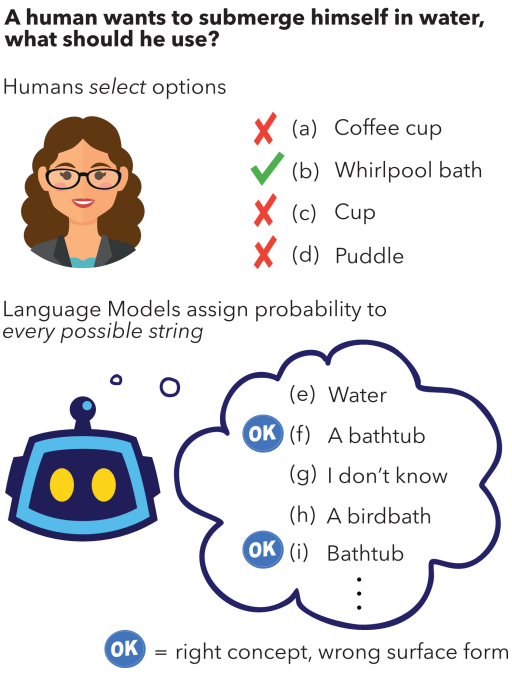

Figure 3.1: While humans select from given options, language models implicitly assign probability to every possible string. This creates surface form competition between different strings that represent the same concept. Example from CommonsenseQA.

PMI_{DC} factors out the probability of a specific surface form, by instead computing how much more probable a hypothesis is when conditioned on a premise. We use a *domain premise* string to estimate the *unconditional* probability of a hypothesis in a given domain. On CommonsenseQA, for example, we compute the probability of each answer option immediately following the string "? the answer is:", and then divide the *conditional* probability by this estimate to calculate PMI_{DC}. This scaling factor reweighs answer scores according to the surface form competition that is inherent to the domain or task, e.g. completions of the domain premise that are just inherently unlikely will be upweighted more. This allows us to directly measure how much an answer tells us about the question and vice versa (mutual information is symmetric, see §3.2). Valid hypotheses no longer need to compete with each other: both "Whirlpool bath" and "Bathtub " will be considered reasonable answers to the question, and so both will attain a high score.

Extensive experiments show that PMI_{DC} consistently outperforms raw, normalized, and calibrated probability scoring methods on zero-shot multiple choice for more than a dozen datasets and it does so for every model in the GPT-2 and GPT-3 families (§3.3); this holds true across different possible prompts and in preliminary few-shot experiments as well. To better explain these gains, we use the distinct structure of the COPA dataset (Roemmele, Bejan, and Gordon, 2011) to remove surface form competition entirely, showing that all methods perform well in this idealized setting (§3.4). Additionally, we analyze the only three datasets where PMI_{DC} does worse than other methods and put forward a hypothesis for why normalizing log probabilities works better than raw probabilities (§3.5). We conclude with a discussion of how generative models should be used for selection tasks (§3.6).

3.1 BACKGROUND AND RELATED WORK

ZERO-SHOT VS. FEW-SHOT Zero-shot inference has long been of interest in NLP, Computer Vision, and ML in general (Guadarrama et al., 2013; Romera-Paredes and Torr, 2015; Socher et al., 2013). However, Radford et al. (2019) popularized the notion that language models have many zero-shot capabilities that can be discovered simply by prompting the model, e.g., placing "TL;DR" (internet slang for Too Long; Didn't Read) at the end of a passage causes the model to generate a summary. Efficiently constructing the right prompt for a given task is difficult and has become an active area of research (Jiang et al., 2020a; b; Lu et al., 2021; Reynolds and McDonell, 2021; Shin et al., 2020).

Brown et al. (2020) demonstrated that few-shot learning without fine-tuning is possible with very large language models. Contemporary work has shown it is possible to get smaller models to exhibit few-shot learning behavior using fine-tuning (Gao, Fisch, and Chen, 2021; Hambardzumyan, Khachatrian, and May, 2021; Schick and Schütze, 2020a; b; c; Shin et al., 2020), an intermediate learning phase (Ye, Lin, and Ren, 2021a), or calibration (Zhao et al., 2021), though most assume access to a validation set (Perez, Kiela, and Cho, 2021). Recent work suggests it may be possible to finetune

language models in order to improve their zero-shot and few-shot capabilities on a large swathe of tasks (Wei et al., 2021; Zhong et al., 2021).

SURFACE FORM COMPETITION When applying generative models to multiple choice problems, simply choosing the *highest probability* answer becomes problematic due to different valid surface forms competing for probability. Indeed, recent work in question answering has demonstrated the importance of considering all multiple choice options together (Khashabi et al., 2020), rather than independently assigning each answer a score and simply choosing the highest. This is a difficult strategy to adapt to left-to-right generative language models, which implicitly choose between *all* possible strings. Using unsupervised language models pretrained on relatively expansive corpora exacerbates surface form competition because such language models generate a much wider distribution than a given question answering dataset contains.

"What is the most populous nation in North America?" Posed with this question, a language model such as GPT-3 can generate a correct response such as "USA", "United States", or "United States of America" with high probability. While correct strings like this all contribute to the probability of a correct generation, they may have vastly different probabilities: a common string "United States" will be much more likely than rarer forms like "U.S. of A.". In generative scenarios, as long as most of the probability mass goes to valid strings the generation is likely to be valid. This is not the case for multiple choice problems. Given two options, e.g., "USA" and "Canada", GPT-3 will choose the correct answer by probability. However, if we substitute out "USA" for "U.S. of A.", GPT-3 will assign higher probability to "Canada", a less likely answer conceptually, but a much more likely surface form. Beyond this, incorrect generic answers such as "I don't know" are often assigned high probability, relegating the desired answers to the tail of the distribution where softmax is poorly calibrated (Holtzman et al., 2020).

PMI Work in dialogue has used PMI to promote diversity (Li et al., 2016a; Mou et al., 2016; Tang et al., 2019; Yao et al., 2017; Zhou et al., 2019). Recently, Brown et al. (2020) used a scoring function resembling PMI_{DC} for zero-shot question answering, though they only use the string "A:" as a prompt for the unconditional probability estimate, whereas we use a task-specific domain premise (see §3.2 for details). Furthermore, Brown et al. (2020) only report this scoring method on three datasets (ARC, OpenBookQA, and RACE, included here) out of the more than 20 tested and do not compare scores with their standard method, averaging log-likelihoods (Avg in this work). In contrast, we report a comprehensive comparison on GPT-3 and GPT-2, as well as shedding light on the underlying issue of surface form competition in §3.4.

CONTEXTUAL CALIBRATION Recently, Zhao et al. (2021) describe a new method for **calibrating** the probabilities of an LM using a learned affine transformation. Though geared towards few-shot learning, the authors devise a clever means of using "content free inputs" for zero-shot learning. Zhao et al. (2021) calibrate for three forms of bias: (1) majority label bias, (2) recency bias, and (3) common token bias. PMI_{DC} directly compensates for common token bias by dividing by the domain conditional probability

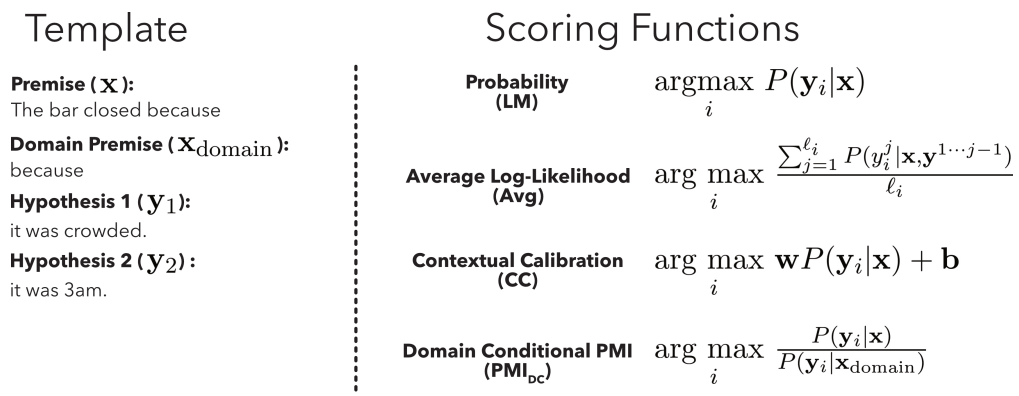

Figure 3.2: An example from COPA () with the template we use as well as the scoring functions we test. LM returns the highest probability option, while Avg length-normalizes log-likelihoods and chooses the highest option. PMI_{DC} is a measurement of the mutual information between hypothesis and premise, intuitively how much $\mathbf{x}$ explains $\mathbf{y}_i$ and vice versa. CC is an affine transform of LM, where $\mathbf{w}$ and $\mathbf{b}$ are averaged over solutions that cause "content free inputs" to yield uniform scores over a given label set, see Zhao et al. (2021).

of each answer, and performs superior to contextual calibration (CC) in the majority of cases.

PROMPT SENSITIVITY Recent work highlights LM sensitivity to *inputs*, and proposes to consider paraphrases of the prompt to overcome this (Davison, Feldman, and Rush, 2019; Jiang et al., 2020a), as well as noting that certain trigger tokens (Shin et al., 2020) can strongly effect the output of such models. In this work, we focus on the surface form of possible *outputs*, but do also analyze robustness to different prompts in §3.3.4.

INTERPRETING LANGUAGE MODELS Language models tend to model selectional preferences and thematic fit (Erk, Padó, and Padó, 2010; Pantel et al., 2007) rather than semantic plausibility (Wang, Durrett, and Erk, 2018). Probability, possibility and plausibility are distinct (Helm, 2006), but reporting bias (Gordon and Van Durme, 2013) means that language models only model what people are likely to write (on websites that are easily crawled). PMI_{DC} aims to adjust for these challenges to better measure the underlying agreement between language models and human judgements, but of course is still subject to the limits and biases of the language model used.

3.2 ZERO-SHOT SCORING STRATEGIES

This chapter does not define any new modeling or finetuning methods. Rather, we propose the broad use of PMI_{DC} scoring for any given model and prompt. PMI_{DC} compensates for the fact that different correct answers compete for probability, even though only one will be listed as the correct multiple choice option.

We begin by describing the two most common methods currently in use.

3.2.1 *Standard Methods*

Our first baseline is simply selecting the highest-probability option, e.g., baselines in Zhao et al. (2021) and Jiang et al. (2020), which we refer to as LM. Given a prompt $\mathbf{x}$ (e.g. "The bar closed") and a set of possible answers $\mathbf{y}_1, \cdots, \mathbf{y}_n$ (e.g. "it was crowded.", "it was 3 AM."), LM is defined:

$$\arg\max_i P(\mathbf{y}_i|\mathbf{x}). \tag{3.1}$$

However, using *length normalized* log-likelihoods (Brown et al., 2020) has become standard due to its superior performance, and is also commonly used in generation (Mao et al., 2019; Oluwatobi and Mueller, 2020). For causal language models, e.g., GPT-2 and GPT-3, Equation 3.1 can be decomposed:

$$P(\mathbf{y}_i|\mathbf{x}) = \prod_{j=1}^{\ell_i} P(y_i^j|\mathbf{x}, y_i^1, \cdots, y_i^{j-1})$$

where y_i^j is the jth token of $\mathbf{y}_i$ and ℓ_i is the number of tokens in $\mathbf{y}_i$. The AVG strategy can thus be defined as:

$$\arg\max_i \frac{\sum_{j=1}^{\ell_i} \log P(y_i^j|\mathbf{x}, \mathbf{y}^{1\cdots j-1})}{\ell_i}.$$

3.2.2 *Domain Conditional PMI*

Our core claim is that direct probability is not an adequate zero-shot scoring function due to surface form competition. A natural solution is to factor out the probability of specific surface forms, which is what Pointwise Mutual Information (PMI) does:

$$\text{PMI}(\mathbf{x}, \mathbf{y}) = \log \frac{P(\mathbf{y}|\mathbf{x})}{P(\mathbf{y})} = \log \frac{P(\mathbf{x}|\mathbf{y})}{P(\mathbf{x})}. \tag{3.2}$$

In effect, this is how much more likely the hypothesis ("it was 3 AM.") becomes given the premise ("The bar closed because"), see Figure 3.2 for the full example. In a multiple-choice setting—where the premise $\mathbf{x}$ does not change across hypotheses—this is proportional to $P(\mathbf{x}|\mathbf{y})$, i.e. the probability of the *premise* given the *hypothesis*. We call this scoring-by-premise and it is the reverse of LM, $P(\mathbf{y}|\mathbf{x})$. We use scoring-by-premise to show the presence of surface form competition in §3.4.

While Equation 3.2 estimates how related premise $\mathbf{x}$ is to hypothesis $\mathbf{y}$ in general, we found that estimates of $P(\mathbf{y})$ vary wildly. GPT-2 and GPT-3 are not trained to produce unconditional estimates of document excerpts, an issue which is exacerbated by the fact that many possible answers are extremely rare in a large scrape of public web pages. This causes the unconditional probability of such answers to be poorly calibrated for the purposes of a given task.

We are specifically trying to measure $P(\mathbf{y})$ in a given domain, e.g., for the "because" relation in our running example, shown in Figures 3.2 & 3.3. To quantify this, we propose *Domain Conditional* PMI:

$$\begin{aligned}\text{PMI}_{\text{DC}}(\mathbf{x}, \mathbf{y}, \text{domain}) &= \frac{P(\mathbf{y}|\mathbf{x}, \text{domain})}{P(\mathbf{y}|\text{domain})} \\ &= \frac{P(\mathbf{y}|\mathbf{x}, \text{domain})}{P(\mathbf{y}|\mathbf{x}_{\text{domain}})}\end{aligned}$$

or how much $\mathbf{x}$ tells us about $\mathbf{y}$ within a domain.

Typically, $P(\mathbf{y}|\mathbf{x}, \text{domain}) = P(\mathbf{y}|\mathbf{x})$ because the premise $\mathbf{x}$ typically implies the domain, e.g., "The bar closed because" sets the model up to predict an independent clause that is the cause of some event, without further representation of the domain. In order to estimate $P(\mathbf{y}|\text{domain})$—the probability of seeing hypothesis $\mathbf{y}$ in a given domain—we use a short domain-relevant string $\mathbf{x}_{\text{domain}}$, which we call a "domain premise", usually just the ending of the conditional premise $\mathbf{x}$. For example, to predict a causal relation like in Figure 3.2 we use $\mathbf{x}_{\text{domain}}$ = "because" and thus divide by $P(\mathbf{y}|\text{because})$–how likely y is to be a "cause" .

3.2.3 *Non-standard Baselines*

UNCONDITIONAL We also compare to the unconditional (in-domain) estimate as a scoring function:

$$\arg\max_i P(\mathbf{y}_i|\mathbf{x}_{\text{domain}}).$$

We refer to this as UNC. It ignores the premise completely, only using a domain premise $\mathbf{x}_{\text{domain}}$ (e.g., using $P(\mathbf{y}|\text{because})$ as the score). Yet, it is sometimes competitive, for instance on BoolQ (Clark et al., 2019). UNC is a sanity check on whether zero-shot inference is actually using the information in the question to good effect.

CONTEXTUAL CALIBRATION Finally, we compare to the reported zero-shot numbers of Zhao et al. (2021). *Contextual Calibration* adjusts LM with an affine transform to make a closed set of answers equally likely in the absence of evidence. Contextual Calibration thus requires computing matrices $\mathbf{w}$ and $\mathbf{b}$ for a number of "content free inputs" and then averaging these weights, see Zhao et al. (2021) for details. In contrast, PMI_{DC} requires nothing but a human-written template (as all zero-shot methods do, including Contextual Calibration), can be computed as the difference of two log probabilities, and is naturally applicable to datasets where the set of valid answers varies between questions.

Multiple Choice Accuracy on GPT-3

Params.	2.7B					6.7B				13B				175B				
	Unc	LM	Avg	PMI_{DC}	CC	Unc	LM	Avg	PMI_{DC}	Unc	LM	Avg	PMI_{DC}	Unc	LM	Avg	PMI_{DC}	CC
COPA	54.8	68.4	68.4	**74.4**	-	56.4	75.8	73.6	**77.0**	56.6	79.2	77.8	**84.2**	56.0	85.2	82.8	**89.2**	-
SC	50.9	66.0	68.3	**73.1**	-	51.4	70.2	73.3	**76.8**	52.0	74.1	77.8	**79.9**	51.9	79.3	83.1	**84.0**	-
HS	31.1	34.5	**41.4**	34.2	-	34.7	40.8	**53.5**	40.0	38.8	48.8	**66.2**	45.8	43.5	57.6	**77.2**	53.5	-
R-M	22.4	37.8	42.4	**42.6**	-	21.2	43.3	45.9	**48.5**	22.9	49.6	50.6	**51.3**	22.5	55.7	**56.4**	55.7	-
R-H	21.4	30.3	32.7	**36.0**	-	22.0	34.8	36.8	**39.8**	22.9	38.2	39.2	**42.1**	22.2	42.4	43.3	**43.7**	-
ARC-E	31.6	**50.4**	44.7	44.7	-	33.5	**58.2**	52.3	51.5	33.8	**66.2**	59.7	57.7	36.2	**73.5**	67.0	63.3	-
ARC-C	21.1	21.6	25.5	**30.5**	-	21.8	26.8	29.8	**33.0**	22.3	32.1	34.3	**38.5**	22.6	40.2	43.2	**45.5**	-
OBQA	10.0	17.2	27.2	**42.8**	-	11.4	22.4	35.4	**48.0**	10.4	28.2	41.2	**50.4**	10.6	33.2	43.8	**58.0**	-
CQA	15.9	33.2	36.0	**44.7**	-	17.4	40.0	42.9	**50.3**	16.4	48.8	47.9	**58.5**	16.3	61.0	57.4	**66.7**	-
BQ	**62.2**	58.5	58.5	53.5	-	37.8	**61.0**	**61.0**	**61.0**	**62.2**	61.1	61.1	60.3	37.8	62.5	62.5	**64.0**	-
RTE	47.3	48.7	48.7	**51.6**	49.5	52.7	**55.2**	**55.2**	48.7	52.7	52.7	52.7	**54.9**	47.3	56.0	56.0	**64.3**	57.8
CB	08.9	51.8	51.8	**57.1**	50.0	08.9	33.9	33.9	**39.3**	08.9	**51.8**	**51.8**	50.0	08.9	48.2	48.2	**50.0**	48.2
SST-2	49.9	53.7	53.76	**72.3**	71.4	49.9	54.5	54.5	**80.0**	49.9	69.0	69.0	**81.0**	49.9	63.6	63.6	71.4	**75.8**
SST-5	18.1	20.0	20.4	**23.5**	-	18.1	27.8	22.7	**32.0**	18.1	18.6	**29.6**	**19.1**	17.6	27.0	27.3	**29.6**	-
AGN	25.0	69.0	69.0	**67.9**	63.2	25.0	**64.2**	**64.2**	57.4	25.0	69.8	69.8	**70.3**	25.0	**75.4**	**75.4**	74.7	73.9
TREC	13.0	29.4	19.2	**57.2**	38.8	22.6	30.2	22.8	**61.6**	22.6	**34.0**	21.4	32.4	22.6	47.2	25.4	**58.4**	57.4

Table 3.1: Comparison of scoring algorithms when using GPT-3 for zero-shot inference on multiple choice questions.

3.3 MULTIPLE CHOICE EXPERIMENTS

3.3.1 *Setup*

We use GPT-2 via the HuggingFace Transformers library (Wolf et al., 2020) and GPT-3 via OpenAI's beta API.[1] We do not finetune any models, nor do we alter their output.

3.3.2 *Datasets*

We report results on 16 splits of 13 datasets, and briefly describe each dataset here.

CONTINUATION These datasets require the model to select a continuation to previous text, making them a natural way to test language models. Choice of Plausible Alternatives (COPA) (Roemmele, Bejan, and Gordon, 2011) asks for cause and effect relationships, as shown in Figure 3.2. StoryCloze (SC) (Mostafazadeh et al., 2017) gives the model a choice between two alternative endings to 5 sentence stories. Finally, HellaSwag (HS) (Zellers et al., 2019a) uses GPT-2 to generate, BERT to filter, and crowd workers to verify possible continuations to a passage. Following previous work (Brown et al., 2020) we report development set numbers for COPA and HS.

1 https://beta.openai.com/

Multiple Choice Accuracy on GPT-2

Params.	125M				350M				760M				1.6B				
	Unc	LM	Avg	PMI_{DC}	Unc	LM	Avg	PMI_{DC}	Unc	LM	Avg	PMI_{DC}	Unc	LM	Avg	PMI_{DC}	CC
COPA	0.564	0.610	**0.632**	0.628	0.558	0.670	0.660	**0.700**	0.556	0.698	0.676	**0.694**	0.560	0.690	0.684	**0.716**	-
SC	0.495	0.600	0.615	**0.670**	0.489	0.630	0.667	**0.716**	0.503	0.661	0.688	**0.734**	0.512	0.676	0.715	**0.763**	-
HS	0.271	0.286	**0.295**	0.291	0.298	0.322	**0.376**	0.328	0.309	0.350	**0.432**	0.351	0.331	0.384	**0.489**	0.378	-
R-M	0.222	0.361	0.406	**0.409**	0.213	0.387	0.420	**0.424**	0.214	0.393	**0.439**	**0.439**	0.223	0.415	0.446	**0.447**	-
R-H	0.209	0.275	0.310	**0.344**	0.215	0.304	0.326	**0.363**	0.215	0.318	0.345	**0.383**	0.219	0.330	0.357	**0.391**	-
ARC-E	0.313	**0.429**	0.378	0.393	0.327	**0.494**	0.434	0.424	0.334	**0.527**	0.467	0.470	0.334	**0.562**	0.496	0.499	-
ARC-C	0.198	0.201	0.235	**0.282**	0.197	0.228	0.254	**0.286**	0.221	0.231	0.266	**0.316**	0.211	0.252	0.279	**0.338**	-
OBQA	0.11	0.164	0.272	**0.324**	0.108	0.186	0.302	**0.386**	0.108	0.194	0.296	**0.432**	0.114	0.224	0.348	**0.460**	-
CQA	0.170	0.255	0.307	**0.364**	0.165	0.309	0.352	**0.418**	0.170	0.333	0.368	**0.445**	0.171	0.386	0.385	**0.478**	-
BQ	**0.622**	0.588	0.588	0.511	**0.622**	0.608	0.608	0.497	**0.622**	0.580	0.580	0.467	**0.622**	0.563	0.563	0.495	-
RTE	**0.527**	0.516	0.516	0.498	0.473	0.531	0.531	**0.549**	0.473	0.531	0.531	**0.542**	0.473	0.477	0.477	**0.534**	0.485
CB	0.089	0.482	0.482	**0.500**	0.089	**0.500**	**0.500**	**0.500**	0.089	0.482	0.482	**0.500**	0.089	**0.500**	**0.500**	**0.500**	0.179
SST-2	0.499	0.636	0.636	**0.671**	0.499	0.802	0.802	**0.862**	0.499	0.770	0.770	**0.856**	0.499	0.840	0.840	**0.875**	0.820
SST-5	0.181	0.274	0.244	**0.300**	0.176	0.185	0.272	**0.393**	0.176	0.203	**0.267**	0.220	0.176	0.304	0.291	**0.408**	-
AGN	0.250	0.574	0.574	**0.630**	0.250	0.643	0.643	**0.644**	0.250	0.607	0.607	**0.641**	0.250	0.648	0.648	**0.654**	0.600
TREC	0.226	0.230	0.144	**0.364**	0.226	**0.288**	0.122	0.216	0.226	0.228	0.226	**0.440**	0.226	0.228	0.240	0.328	**0.340**

Table 3.2: Comparison of scoring algorithms when using GPT-2 for zero-shot inference on multiple choice questions.

QUESTION ANSWERING RACE-M & -H (R-M & R-H) (Lai et al., 2017) are both drawn from English exams given in China, the former being given to Middle Schoolers and the latter to High Schoolers. Similarly, ARC Easy & Challenge (ARC-E & ARC-C) (Clark et al., 2018) are standardized tests described as "natural, grade-school science questions," with the "Easy" split found to be solvable by either a retrieval or word co-occurrence system, and the rest of the questions put in the "Challenge" split. Open Book Question Answering (OBQA) (Mihaylov et al., 2018) is similar to both of these, but was derived using (and intended to be tested with) a knowledge source (or "book") available; we do not make use of the given knowledge source, following Brown et al. (2020). Finally, CommonsenseQA (CQA) (Talmor et al., 2019) leverages ConceptNet (Speer, Chin, and Havasi, 2017) to encourage crowd workers to write questions with challenging distractors. We report development set numbers on CQA because their test set is not public.

OPEN SET VS. CLOSED SET DATASETS The above datasets are all "open set" in that multiple choice answers may be any string. Below we describe "closed set" datasets with a fixed set of answers.

BOOLEAN QUESTION ANSWERING BoolQ (BQ) (Clark et al., 2019) poses yes/no (i.e. Boolean) questions based on a multi-sentence passage.

ENTAILMENT Entailment datasets focus on the question of whether a hypothesis sentence B is entailed by a premise sentence A. Recognizing Textual Entailment (RTE)

Method	Unc	LM	Avg	PMI_{DC}	CC
125M	12.50	6.25	12.50	**68.75**	-
350M	6.25	18.75	12.50	**68.75**	-
760M	6.25	6.25	12.50	**75.00**	-
1.6B	6.25	12.50	12.50	**80.00**	20.00
2.7B	6.25	6.25	6.25	**86.66**	0.00
6.7B	6.25	25.00	25.00	**75.00**	-
13B	6.25	18.75	18.75	**68.75**	-
175B	6.25	12.50	18.75	**62.50**	6.25

Table 3.3: Percentage of datasets that given methods produce the best score or tie with other methods, aggregated over each model size. The first four rows use GPT-2 (full data available in the Appendix), while the final four rows use GPT-3 and summarize data from Table 3.1. Since ties are included, rows sometimes sum to more than 100. CC is only measured on the 5 datasets we use where Zhao et al. (2021) also report accuracies.

(Dagan, Glickman, and Magnini, 2005) requires predicting an "entailment" or "contradiction" label while Commitment Bank (CB) (De Marneffe, Simons, and Tonhauser, 2019) adds a "neutral" label. Following previous work (Brown et al., 2020) we report development set numbers for both RTE and CB.

TEXT CLASSIFICATION We consider three more complex classification datasets: SST-2 & -5 (Socher et al., 2013) for various granularities of sentiment classification, AG's News (Zhang, Zhao, and LeCun, 2015) (AGN) for topic classification, and TREC (Li and Roth, 2002) for question classification.

3.3.3 *Results*

We report zero-shot results for GPT-3 in Table 3.1 and GPT-2 results in Table 3.2 . A summarized view is shown in Table 3.3, which aggregates the percentage of splits where a given method achieves the best score or ties for first-place. In this summarized view it is clear that PMI_{DC} consistently outperforms other scoring methods when assessed over a variety of datasets. The smallest margin (in number of datasets won or tied) between PMI_{DC} and the best competing method is on GPT-3 175B with Avg, but that margin is over 40 percentage points. This does not imply that PMI_{DC} is *always* better or that it will be better by a large margin, though it often is. It does suggest that PMI_{DC} is a significantly better bet on a new dataset.

3.3.4 *Robustness*

To verify that these trends hold across different prompts, we report the mean and standard deviation over the fifteen different prompts considered in (Zhao et al., 2021) for SST-2. Table 3.4 shows, PMI_{DC} always maintains the highest mean, often by a hefty margin. Scores are lower than in Table 3.1 because many of the prompts used are optimized for few-shot rather than zero-shot scoring.

Prompt Robustness on SST-2

Method	Unc	LM	PMI_{DC}
125M	49.9 $_{0}$	56.8 $_{7.3}$	**58.8** $_{7.6}$
350M	49.9 $_{0}$	58.0 $_{11.3}$	**60.3** $_{11.4}$
760M	49.9 $_{0}$	57.0 $_{9.2}$	**67.7** $_{13.4}$
1.6B	49.9 $_{0}$	57.3 $_{8.2}$	**69.8** $_{13.3}$
2.7B	49.9 $_{0}$	56.1 $_{9.0}$	**66.2** $_{15.7}$
6.7B	49.9 $_{0}$	59.5 $_{10.7}$	**67.9** $_{13.6}$
13B	49.9 $_{0}$	63.0 $_{14.9}$	**71.7** $_{16.1}$
175B	49.9 $_{0}$	72.5 $_{15.7}$	**74.8** $_{14.0}$

Table 3.4: The mean and standard deviations over the 15 templates considered for SST-2 in (). Avg is excluded, as it is equivalent to LM since all the given templates use single-token answers.

4-shot Inference Results

	SST-2			CQA			
Method	Unc	LM	PMI_{DC}	Unc	LM	Avg	PMI_{DC}
125M	49.9 $_{0}$	63.6 $_{7.4}$	**71.7** $_{5.1}$	15.5 $_{0}$	29.9 $_{1.6}$	32.7 $_{1.4}$	**38.3** $_{1.7}$
350M	49.9 $_{0}$	76.3 $_{13.8}$	**76.4** $_{8.1}$	16.5 $_{0}$	37.6 $_{2.3}$	40.4 $_{2.3}$	**45.7** $_{2.4}$
760M	49.9 $_{0}$	85.9 $_{7.2}$	**87.1** $_{3.0}$	16.1 $_{0}$	41.5 $_{2.6}$	42.4 $_{2.5}$	**47.0** $_{1.5}$
1.6B	49.9 $_{0}$	85.4 $_{1.7}$	**89.4** $_{4.0}$	16.0 $_{0}$	46.2 $_{1.5}$	47.7 $_{1.9}$	**52.3** $_{2.1}$
2.7B	49.9 $_{0}$	**88.1** $_{4.9}$	87.7 $_{5.5}$	16.6 $_{0}$	43.0 $_{1.7}$	45.6 $_{1.9}$	**50.4** $_{1.1}$
6.7B	49.9 $_{0}$	**92.9** $_{2.1}$	79.8 $_{6.9}$	16.9 $_{0}$	52.3 $_{1.4}$	53.4 $_{1.0}$	**56.5** $_{1.6}$
13B	49.9 $_{0}$	85.4 $_{9.0}$	**86.9** $_{7.5}$	16.7 $_{0}$	58.4 $_{2.0}$	59.3 $_{1.5}$	**63.4** $_{1.4}$
175B	49.9 $_{0}$	89.9 $_{5.5}$	**95.5** $_{0.7}$	16.5 $_{0}$	69.1 $_{1.9}$	69.4 $_{0.8}$	**72.0** $_{0.9}$

Table 3.5: The mean and standard deviation for 5 randomly sampled sets of 4 examples used for few-shot inference. We include a closed answer dataset (SST-2) and an open answer dataset (CQA). For SST-2 Avg is equivalent to LM due to using single-token answers.

3.3.5 *Few-shot*

While our focus in this chapter is on zero-shot scoring, PMI_{DC} is just as applicable to few-shot scenarios. In Table 3.5 we report 4-shot results on one closed set dataset (SST-2) and one open set dataset (CQA). We show the mean of 5 randomly sampled sets of 4 examples that are used to prime the model for the task, along with standard deviations. The overall trend on both datasets clearly favors PMI_{DC}, though LM is superior for two models on SST-2.

3.4 REMOVING SURFACE FORM COMPETITION

What if we used the probability of the *premise* given the *hypothesis*, $P(\mathbf{x}|\mathbf{y}_i)$, instead? While we are still measuring the probability of a surface form (e.g. "the bar closed."), it is the *same* surface form across different options ("It was crowded so", "It was 3 AM so"),

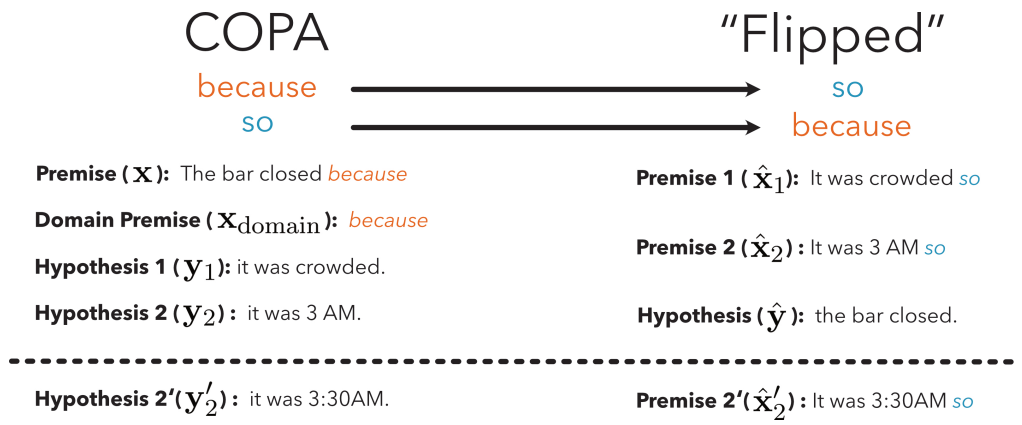

Figure 3.3: In §3.4 we experiment with with flipping the premise and hypothesis so that the highest probability *premise* is chosen as the answer, i.e. scoring-by-premise. The transformation above the dashed line shows the experimental setup used in §3.4.1, while the extra distractor below the dashed line is used for illustrative purposes in §3.4.2.

eliminating the surface form competition. $\mathbf{y}_i$ and $\mathbf{y}'_i$ can now both attain high scores if they are both correct answers, by causing $\mathbf{x}$ to be likely. We call this scoring-by-premise.

Causal language models like GPT-3 cannot measure this directly, because they are only capable of conditioning on past tokens to predict future tokens. We exploit the structure of the COPA dataset to create "COPA Flipped" via a simple transformation, shown in Figure 3.3. COPA consists of cause and effect pairs (CAUSE *so* EFFECT, and EFFECT *because* CAUSE). In the original dataset, whatever comes second (either CAUSE or EFFECT) has two options that a model must choose between. These can be reversed by switching CAUSE and EFFECT, then substituting the natural inverse relation ("because"→"so" and "so"→"because").

3.4.1 *Results*

Table 3.6 shows scores on COPA and COPA Flipped side-by-side. On COPA Flipped everything except UNC produces the *exact* same result. This is because flipping the hypothesis and premise means that it's the *context* that changes and not the *continuation*. LM, AVG, and PMI_{DC} only differ from each other over different continuations, not over different contexts for the same continuation.

On COPA Flipped all methods generally perform similarly to PMI_{DC} on the unflipped version. This is because surface form competition has been eradicated: we are measuring how well different prefixes condition a model to predict a fixed continuation rather than which continuation is highest probability. Unlike LM, where different answers compete for probability, in COPA Flipped it only matters how likely each answer can make the question. This is not subject to surface form competition because there is only one string being so scored, so it is not competing with any other strings for probability mass.

Removing Surface Form Competition

	COPA				COPA Flipped			
Method	Unc	LM	Avg	PMI_{DC}	Unc	LM	Avg	PMI_{DC}
125M	56.4	61.0	63.2	62.8	50.0	63.2	63.2	63.2
350M	55.8	67.0	66.0	70.0	50.0	66.4	66.4	66.4
760M	55.6	69.8	67.6	69.4	50.0	70.8	70.8	70.8
1.6B	56.0	69.0	68.4	71.6	50.0	73.0	73.0	73.0
2.7B	54.8	68.4	68.4	74.4	50.0	68.4	68.4	68.4
6.7B	56.4	75.8	73.6	77.0	50.0	76.8	76.8	76.8
13B	56.6	79.2	77.8	84.2	50.0	79.0	79.0	79.0
175B	56.0	85.2	82.8	89.2	50.0	83.6	83.6	83.6

Table 3.6: LM does better on COPA Flipped than COPA because surface form competition is removed when scoring-by-premise, see §3.4. Methods that don't directly adjust for competing surface forms (LM and Avg) have the same score as PMI_{DC} on COPA Flipped.

Not all datasets are so easily flippable, so manually flipping individual questions to remove surface form competition is not a generally applicable strategy. Luckily, PMI_{DC} is symmetric:

$$\begin{aligned}
&\arg\max_i \frac{P(\mathbf{y}_i|\mathbf{x}, \text{domain})}{P(\mathbf{y}_i|\text{domain})} \\
&= \arg\max_i \frac{P(\mathbf{x}|\mathbf{y}_i, \text{domain})}{P(\mathbf{x}|\text{domain})} \\
&= \arg\max_i P(\mathbf{x}|\mathbf{y}_i, \text{domain})
\end{aligned}$$

In theory, the answer selected by PMI_{DC} should be the same between COPA and COPA Flipped as PMI is symmetric, though we expect some differences due to "so" and "because" not being perfect inverses and shuffled references. Thus, PMI_{DC} does better on COPA than COPA Flipped, likely due to more natural phrasing in the original dataset.

These results suggest that surface form competition is the primary cause of the depressed performance of LM and Avg in comparison to PMI_{DC}.

3.4.2 *In-depth Example*

SCORING-BY-PREMISE IMPROVES LM Figure 3.3 shows an example of transforming one question from COPA to COPA Flipped. In the example depicted, when we use GPT-3 to calculate P, we get:

$$P(\mathbf{y}_1|\mathbf{x}) > P(\mathbf{y}_2|\mathbf{x})$$

which is wrong, since bars usually close at fixed, late-night closing times, rather than because of being overcrowded. However we also find that

$$\begin{aligned}
P(\hat{\mathbf{y}}|\hat{\mathbf{x}}_2) &> P(\hat{\mathbf{y}}|\hat{\mathbf{x}}_1) \\
\frac{P(\mathbf{y}_2|\mathbf{x})}{P(\mathbf{y}_2|\mathbf{x}_{\text{domain}})} &> \frac{P(\mathbf{y}_1|\mathbf{x})}{P(\mathbf{y}_1|\mathbf{x}_{\text{domain}})}
\end{aligned}$$

indicating that scoring-by-premise causes the right answer to be selected and that PMI_{DC} successfully simulates scoring by premise in this example.

STABILITY OVER VALID ANSWERS To see how scoring-by-premise allows multiple correct options to achieve high scores, consider the slightly perturbed $\mathbf{y}_2'$ and $\hat{\mathbf{x}}_2'$ in Figure 3.3. The inequalities shown above still hold when substituting $\mathbf{y}_2 \rightarrow \mathbf{y}_2'$ and $\hat{\mathbf{x}}_2 \rightarrow \hat{\mathbf{x}}_2'$:

$$P(\mathbf{y}_1|\mathbf{x}) > P(\mathbf{y}_2'|\mathbf{x})$$
$$P(\hat{\mathbf{y}}|\hat{\mathbf{x}}_2') > P(\hat{\mathbf{y}}|\hat{\mathbf{x}}_1)$$
$$\frac{P(\mathbf{y}_2'|\mathbf{x})}{P(\mathbf{y}_2'|\mathbf{x}_{\text{domain}})} > \frac{P(\mathbf{y}_1|\mathbf{x})}{P(\mathbf{y}_1|\mathbf{x}_{\text{domain}})}$$

with the key difference that the conditional probability of $\mathbf{y}_2'$ is much lower:

$$\log P(\mathbf{y}_2|\mathbf{x}) \approx -16$$
$$\log P(\mathbf{y}_2'|\mathbf{x}) \approx -20$$

This is undesirable, as both $\mathbf{y}_2$ and $\mathbf{y}_2'$ are correct answers with similar meanings. Yet, when scoring-by-premise the conditional probability of $\hat{\mathbf{y}}$ is stable when substituting $\hat{\mathbf{x}}_2 \rightarrow \hat{\mathbf{x}}_2'$:

$$\log P(\hat{\mathbf{y}}|\hat{\mathbf{x}}_2) \approx -12$$
$$\log P(\hat{\mathbf{y}}|\hat{\mathbf{x}}_2') \approx -12$$

This suggests that eliminating surface form competition allows different correct answers to score well, as they are no longer competing for probability mass. Specifically, "it was 3 AM" and "it was 3:30AM" score wildly differently in COPA but nearly identically in COPA Flipped.

3.5 ANALYSIS

FAILURE CASES There are three datasets where PMI_{DC} does not consistently outperform other methods: HellaSwag, ARC Easy, and BoolQ. Surprisingly, each is dominated by a different method.

HellaSwag is most amenable to Avg. On examination we find that HellaSwag is more focused on the *internal coherence* of the hypotheses, rather than *external coherence*, i.e. how much a premise and hypothesis match. This is likely due to HellaSwag being generated by GPT-2 (Radford et al., 2019c) and filtered with BERT, as it contains relatively on-topic but intrinsically strange hypotheses that humans can distinguish from natural data.

ARC Easy yields the highest scores to LM, i.e., selecting the highest probability option. Clark et al. (2018) note that ARC Easy questions can be solved by a retrieval or word co-occurrence baseline, while examples that were answered incorrectly by both were put into the Challenge split. This suggests a bias towards a priori likely phrases.

Manual inspection reveals many stock answers, e.g., "[clouds are generated when] ocean water evaporates and then condenses in the air," supporting our hypothesis.

Finally, BoolQ, a reading comprehension dataset in which all answers are either "yes" or "no", is best solved by an unconditional baseline. This is because the dataset presents truly complex questions that require more reasoning than GPT-2 or 3 are capable of out of the box. Indeed, none of the methods reported do better than the majority baseline, except PMI_{DC} with the largest GPT-3 model.

WHY DOES LENGTH NORMALIZATION WORK? Past work offers little explanation for why AVG should be a successful strategy, other than the intuition that estimates are strongly length biased and require compensation. Length bias may be caused by the final softmax layer of current language models assigning too much probability mass to irrelevant options at each time-step, as noted in open-ended generation, character-level language modeling, and machine translation (Al-Rfou et al., 2019; Holtzman et al., 2020; Peters, Niculae, and Martins, 2019).

Another (not mutually exclusive) argument is that length normalization may account for *unconditional probability* in a similar way to PMI_{DC}. Length normalization is often measured over Byte Pair Encoding (BPE) tokens (Sennrich, Haddow, and Birch, 2016) and BPE tends to produce vocabularies where most tokens are equally frequent (Wang, Cho, and Gu, 2020). Recent evidence suggests that language is approximately uniformly information dense (Jaeger, 2006; Levy, 2018; Levy and Jaeger, 2007). As such, length in BPE tokens may correspond roughly to a *unigram* estimate of log-probability, supposing that BPE tokens have approximately uniform unigram frequency. The adjustment made by AVG is still somewhat different than PMI_{DC}, (division of log terms rather than subtraction) but could have a similar effect, if length and probability correlate.

3.6 DISCUSSION

Language Models are density estimation functions that assign probability to every possible string, but there are often many strings that could represent a given idea equally well. Our key observation is that a generative model assigning probability to a string that *represents* a certain option isn't equivalent to selecting the *concept* an option corresponds to. We expect surface form competition anywhere that generative models are used where more than one string could represent the same concept.

PMI_{DC} aligns the predictions being made by the model more closely with the actual task posed by multiple choice questions: "choose the hypothesis that explains the premise" rather than "generate the exact surface form of the hypothesis". From this perspective, PMI_{DC} does not go far enough, because the model still cannot consider the given set of options altogether when selecting its choice. This matters when answers interact with each other, e.g., "all of the above".

3.7 SUPPLEMENTARY A

Table 3.7 shows an example of each template used for each dataset.

Type	Dataset	Template
Continuation	COPA	[The man broke his toe]$_{\text{P}}$ [because]$_{\text{DP}}$ [he got a hole in his sock.]$_{\text{UH}}$
		[I tipped the bottle]$_{\text{P}}$ [so]$_{\text{DP}}$ [the liquid in the bottle froze.]$_{\text{UH}}$
	StoryCloze	[Jennifer has a big exam tomorrow. She got so stressed, she pulled an all-nighter. She went into class the next day, weary as can be. Her teacher stated that the test is postponed for next week.]$_{\text{P}}$ [The story continues:]$_{\text{DP}}$ [Jennifer felt bittersweet about it.]$_{\text{UH}}$
	HellaSwag	[A female chef in white uniform shows a stack of baking pans in a large kitchen presenting them. the pans]$_{\text{P}}$ [contain egg yolks and baking soda.]$_{\text{UH}}$
QA	RACE	[There is not enough oil in the world now. As time goes by, it becomes less and less, so what are we going to do when it runs out [...].]$_{\text{P}}$ question: [According to the passage, which of the following statements is true]$_{\text{P}}$[?]$_{\text{DP}}$ answer: [There is more petroleum than we can use now.]$_{\text{UH}}$
	ARC	[What carries oxygen throughout the body?]$_{\text{P}}$ [the answer is:]$_{\text{DP}}$ [red blood cells.]$_{\text{UH}}$
	OBQA	[Which of these would let the most heat travel through?]$_{\text{P}}$ [the answer is:]$_{\text{DP}}$ [a steel spoon in a cafeteria.]$_{\text{UH}}$
	CQA	[Where can I stand on a river to see water falling without getting wet?]$_{\text{P}}$ [the answer is:]$_{\text{DP}}$ [bridge.]$_{\text{UH}}$
Boolean QA	BoolQ	title: [The Sharks have advanced to the Stanley Cup finals once, losing to the Pittsburgh Penguins in 2016 [...]]$_{\text{P}}$ question: [Have the San Jose Sharks won a Stanley Cup?]$_{\text{P}}$ [answer:]$_{\text{DP}}$ [No.]$_{\text{UH}}$
Entailment	RTE	[Time Warner is the world's largest media and Internet company.]$_{\text{P}}$ question: [Time Warner is the world's largest company.]$_{\text{P}}$ [true or false? answer:]$_{\text{DP}}$ [true.]$_{\text{UH}}$
	CB	question: Given that [What fun to hear Artemis laugh. She's such a serious child.]$_{\text{P}}$ Is [I didn't know she had a sense of humor.]$_{\text{P}}$ true, false, or neither? [the answer is:]$_{\text{DP}}$ [true.]$_{\text{UH}}$
Text Classification	SST-2	"[Illuminating if overly talky documentary]$_{\text{P}}$" [[The quote] has a tone that is]$_{\text{DP}}$ [positive.]$_{\text{UH}}$
	SST-5	"[Illuminating if overly talky documentary]$_{\text{P}}$" [[The quote] has a tone that is]$_{\text{DP}}$ [neutral.]$_{\text{UH}}$
	AG's News	title: [Economic growth in Japan slows down as the country experiences a drop in domestic and corporate [...]]$_{\text{P}}$ summary: [Expansion slows in Japan]$_{\text{P}}$ [topic:]$_{\text{DP}}$ [Sports.]$_{\text{UH}}$
	TREC	[Who developed the vaccination against polio?]$_{\text{P}}$ [The answer to this question will be]$_{\text{DP}}$ [a person.]$_{\text{UH}}$

Table 3.7: The templates used for each task, along with an example instance (with a single random candidate answer). Original questions (premises) are colored blue, and original answers (hypotheses) are colored red. Long premises are abbreviated with "[...]". The full premises, conditional hypotheses and domain premises are marked in $[\cdot]_{\text{P}}$, $[\cdot]_{\text{UH}}$, and $[\cdot]_{\text{DP}}$ respectively. For a complete description of our templating methodology.

Part II

MODELING MODELS

4

LEARNING TO WRITE WITH COOPERATIVE DISCRIMINATORS

This section is adapted from Holtzman et al. (2018)

Language models based on Recurrent Neural Networks (RNNs) have brought substantial advancements across a wide range of language tasks (Bahdanau, Cho, and Bengio, 2015b; Chopra, Auli, and Rush, 2016; Jozefowicz et al., 2016). However, when used for long-form text generation, RNNs often lead to degenerate text that is repetitive, self-contradictory, and overly generic, as shown in Figure 4.1.

We propose a unified learning framework that can address several challenges of long-form text generation by composing a committee of discriminators each specializing in a different principle of communication. Starting with an RNN language model, our framework learns to construct a more powerful generator by training a number of discriminative models that can collectively address limitations of the base RNN generator, and then learns how to weigh these discriminators to form the final decoding objective. These "cooperative" discriminators complement each other and the base language model to form a stronger, more global decoding objective.

The design of our discriminators are inspired by Grice's maxims (Grice, Cole, Morgan, et al., 1975) of quantity, quality, relation, and manner. The discriminators learn to encode these qualities through the selection of training data (e.g. distinguishing a true continuation from a randomly sampled one as in §3.2 Relevance Model), which includes generations from partial models (e.g. distinguishing a true continuation from one generated by a language model as in §3.2 Style Model). The system then learns to balance these discriminators by initially weighing them uniformly, then continually updating its weights by comparing the scores the system gives to its own generated continuations and to the reference continuation.

Empirical results (§4.4) demonstrate that our learning framework is highly effective in converting a generic RNN language model into a substantially stronger generator. Human evaluation confirms that language generated by our model is preferred over that of competitive baselines by a large margin in two distinct domains, and significantly enhances the overall coherence, style, and information content of the generated text. Automatic evaluation shows that our system is both less repetitive and more diverse than baselines.

4.1 BACKGROUND

RNN language models learn the conditional probability $P(x_t|x_1, ..., x_{t-1})$ of generating the next word x_t given all previous words. This conditional probability learned by RNNs often assigns higher probability to repetitive, overly generic sentences, as shown in Figure 4.1 and also in Table 4.3. Even gated RNNs such as LSTMs (Hochreiter and Schmidhuber, 1997) and GRUs (Cho et al., 2014) have difficulties in properly

Context: The two guards thudded into the door on the other side and began pounding on it, shouting furiously. Scious raised the large bunch of keys then placed them in a large pocket in his coat. "Come, we have to go," he whispered and moved up the corridor, the guards still hammering on the door. Jared looked around. The table next to the room they had been in was bare.

LM: There was a small table set into the center of the table, and a table with two chairs and a table. A chair sat next to the couch. A large man stood in the doorway of the room, his arms crossed over his chest. "What 's up?" He asked.

L2W: Empty. It looked like some kind of *prison*. There were no windows, no doors, no windows. *No sign of anyone in the house.* No one had ever been here before.

Reference: Only a flagon of wine and a couple of mugs plus a crude dice game. Light flickered in the stone corridor from lanterns on the walls. The place was dank and gloomy, moss in scattered patches on the floor and walls. The corridor ended just beyond the door to their former *prison*. *No one else was about.*

Figure 4.1: Sample generations from an RNN language model (LM) and our system (L2W) conditioning on the context shown on the top. The red, underlined text highlights repetitions, while the *blue, italicized* text highlights details that have a direct semantic parallel in the reference text.

incorporating long-term context due to explaining-away effects (Yu et al., 2017b), diminishing gradients (Pascanu, Mikolov, and Bengio, 2013), and lack of inductive bias for the network to learn discourse structure or global coherence beyond local patterns.

Several methods in the literature attempt to address these issues. Overly simple and generic generation can be improved by length-normalizing the sentence probability (Wu et al., 2016), future cost estimation (Schmaltz, Rush, and Shieber, 2016), or a diversity-boosting objective function (Shao et al., 2017; Vijayakumar et al., 2016). Repetition can be reduced by prohibiting recurrence of the trigrams as a hard rule (Paulus, Xiong, and Socher, 2018). However, such hard constraints do not stop RNNs from repeating through paraphrasing while preventing occasional intentional repetition.

We propose a unified framework to address all these related challenges of long-form text generation by learning to construct a better decoding objective, generalizing over various existing modifications to the decoding objective.

4.2 THE LEARNING FRAMEWORK

We propose a general learning framework for conditional language generation of a sequence $\mathbf{y}$ given a fixed context $\mathbf{x}$. The decoding objective for generation takes the general form

$$f_\lambda(\mathbf{x}, \mathbf{y}) = \log(P_{\text{lm}}(\mathbf{y}|\mathbf{x})) + \sum_k \lambda_k s_k(\mathbf{x}, \mathbf{y}), \tag{4.1}$$

where every s_k is a scoring function. The proposed objective combines the RNN language model probability P_{lm} (§4.2.1) with a set of additional scores $s_k(\mathbf{x}, \mathbf{y})$ produced by discriminatively trained communication models (§4.2.2), which are weighted with learned mixture coefficients λ_k (§4.2.3). When the scores s_k are log probabilities, this corresponds to a Product of Experts (PoE) model (Hinton, 2002).

Generation is performed using beam search (§4.2.4), scoring incomplete candidate generations $\mathbf{y}_{1:i}$ at each time step i. The RNN language model decomposes into per-word probabilities via the chain rule. However, in order to allow for more expressivity over long range context we do not require the discriminative model scores to factorize over the elements of $\mathbf{y}$, addressing a key limitation of RNNs. More specifically, we use an estimated score $s'_k(\mathbf{x}, \mathbf{y}_{1:i})$ that can be computed for any prefix of $\mathbf{y} = \mathbf{y}_{1:n}$ to approximate the objective during beam search, such that $s'_k(\mathbf{x}, \mathbf{y}_{1:n}) = s_k(\mathbf{x}, \mathbf{y})$. To ensure that the training method matches this approximation as closely as possible, scorers are trained to discriminate prefixes of the same length (chosen from a predetermined set of prefix lengths), rather than complete continuations, except for the entailment module as described in §3.2 Entailment Model. The prefix scores are re-estimated at each time-step, rather than accumulated over beam search.

4.2.1 *Base Language Model*

The RNN language model treats the context $\mathbf{x}$ and the continuation $\mathbf{y}$ as a single sequence $\mathbf{s}$:

$$\log P_{\text{lm}}(\mathbf{s}) = \sum_i \log P_{\text{lm}}(\mathbf{s}_i|\mathbf{s}_{1:i-1}). \tag{4.2}$$

4.2.2 *Cooperative Communication Models*

We introduce a set of discriminators, each of which encodes an aspect of proper writing that RNNs usually fail to capture. Each model is trained to discriminate between good and bad generations; we vary the model parameterization and training examples to guide each model to focus on a different aspect of Grice's Maxims. The discriminator scores are interpreted as classification probabilities (scaled with the logistic function where necessary) and interpolated in the objective function as log probabilities.

Let $D = \{(\mathbf{x_1}, \mathbf{y_1}), \ldots (\mathbf{x_n}, \mathbf{y_n})\}$ be the set of training examples for conditional generation. $D_{\mathbf{x}}$ denote all contexts and $D_{\mathbf{y}}$ all continuations. The scoring functions are trained on prefixes of $\mathbf{y}$ to simulate their application to partial continuations at inference time.

In all models the first layer embeds each word w into a 300-dimensional vector $e(w)$ initialized with GloVe (Pennington, Socher, and Manning, 2014) pretrained-embeddings.

Repetition Model

This model addresses the maxim of Quantity by biasing the generator to avoid repetitions. The goal of the repetition discriminator is to learn to distinguish between RNN-generated and gold continuations by exploiting our empirical observation that repetitions are more common in completions generated by RNN language models. However, we do not want to completely eliminate repetition, as words do recur in English.

In order to model natural levels of repetition, a score d_i is computed for each position in the continuation $\mathbf{y}$ based on pairwise cosine similarity between word embeddings within a fixed window of the previous k words, where

$$d_i = \max_{j=i-k \ldots i-1} (\text{CosSim}(e(y_j), e(y_i))), \tag{4.3}$$

such that $d_i = 1$ if y_i is repeated in the window.

The score of the continuation is then defined as

$$s_{\text{rep}}(\mathbf{y}) = \sigma(\mathbf{w}_r^\top \text{RNN}_{\text{rep}}(\mathbf{d})), \tag{4.4}$$

where $\text{RNN}_{\text{rep}}(\mathbf{d})$ is the final state of a unidirectional RNN ran over the similarity scores $\mathbf{d} = d_1 \ldots d_n$ and $\mathbf{w}_r$ is a learned vector. The model is trained to maximize the ranking log likelihood

$$L_{\text{rep}} = \sum_{\substack{(\mathbf{x},\mathbf{y}_g) \in D, \\ \mathbf{y}_s \sim \text{LM}(\mathbf{x})}} \log \sigma(s_{\text{rep}}(\mathbf{y}_g) - s_{\text{rep}}(\mathbf{y}_s)), \tag{4.5}$$

which corresponds to the probability of the gold ending $\mathbf{y}_g$ receiving a higher score than the ending sampled from the RNN language model.

Entailment Model

Judging textual quality can be related to the natural language inference (NLI) task of recognizing textual entailment (Bowman et al., 2015; Dagan, Glickman, and Magnini, 2006): we would like to guide the generator to neither contradict its own past generation (the maxim of Quality) nor state something that readily follows from the context (the maxim of Quantity). The latter case is driven by the RNNs habit of paraphrasing itself during generation.

We train a classifier that takes two sentences a and b as input and predicts the relation between them as either *contradiction*, *entailment* or *neutral*. We use the *neutral* class probability of the sentence pair as discriminator score, in order to discourage both contradiction and entailment. As entailment classifier we use the decomposable

attention model (Parikh et al., 2016), a competitive, parameter-efficient model for entailment classification.[1] The classifier is trained on two large entailment datasets, SNLI (Bowman et al., 2015) and MultiNLI (Williams, Nangia, and Bowman, 2017), which together have more than 940,000 training examples. We train separate models based on the vocabularies of each of the datasets we use for evaluation.

In contrast to our other communication models, this classifier cannot be applied directly to the full context and continuation sequences it is scoring. Instead every completed sentence in the continuation should be scored against all preceding sentences in both the context and continuation.

Let $t(\mathbf{a}, \mathbf{b})$ be the log probability of the neutral class. Let $S(\mathbf{y})$ be the set of complete sentences in $\mathbf{y}$, $S_{\text{last}}(\mathbf{y})$ the last complete sentence, and $S_{\text{init}}(\mathbf{y})$ the sentences before the last complete sentence. We compute the entailment score of $S_{\text{last}}(\mathbf{y})$ against all preceding sentences in $\mathbf{x}$ and $\mathbf{y}$, and use the score of the sentence-pair for which we have the least confidence in a *neutral* classification:

$$s_{\text{entail}}(\mathbf{x}, \mathbf{y}) = min_{\mathbf{a} \in S(\mathbf{x}) \cup S_{\text{init}}(\mathbf{y})} t(\mathbf{a}, S_{\text{last}}(\mathbf{y})). \tag{4.6}$$

Intuitively, we only use complete sentences because the ending of a sentence can easily flip entailment. As a result, we carry over entailment score of the last complete sentence in a generation until the end of the next sentence, in order to maintain the presence of the entailment score in the objective. Note that we check that the current sentence is not directly entailed or contradicted by a previous sentence and not the reverse. [2] In contrast to our other models, the score this model returns only corresponds to a subsequence of the given continuation, as the score is not accumulated across sentences during beam search. Instead the decoder is guided locally to continue complete sentences that are not entailed or contradicted by the previous text.

Relevance Model

The relevance model encodes the maxim of Relation by predicting whether the content of a candidate continuation is relevant to the given context. We train the model to distinguish between true continuations and random continuations sampled from other (human-written) endings in the corpus, conditioned on the given context.

First both the context and continuation sequences are passed through a convolutional layer, followed by maxpooling to obtain vector representations of the sequences:

$$a = \text{maxpool}(\text{conv}_a(e(\mathbf{x}))), \tag{4.7}$$

$$b = \text{maxpool}(\text{conv}_b(e(\mathbf{y}))). \tag{4.8}$$

The goal of maxpooling is to obtain a vector representing the most important semantic information in each dimension.

1 We use the version without intra-sentence attention.

2 If the current sentence entails a previous one it may simply be adding more specific information, for instance: "He hated broccoli. Every time he ate broccoli he was reminded that it was the thing he hated most."

Data: context $\mathbf{x}$, beam size k, sampling temperature t
Result: best continuation
best = None
beam = [$\mathbf{x}$]
for step = 0; step < max_steps; step = step +1 **do**
  next_beam = []
  for candidate in beam **do**
    next_beam.extend(next_k(candidate))
    if termination_score(candidate) > best.score **then**
      best = candidate.append(term)
    end
  end
  for candidate in next_beam **do**
    ▷ score with models
    candidate.score += f_λ(candidate)
  end
  ▷ sample k candidates by score
  beam = sample(next_beam, k, t)
end
if learning **then**
  update λ with gradient descent by comparing best against the gold.
end
return best

Algorithm 1: Inference/Learning in the Learning to Write Framework.

The scoring function is then defined as

$$s_{\text{rel}} = \mathbf{w}_l^T \cdot (a \circ b), \tag{4.9}$$

where element-wise multiplication of the context and continuation vectors will amplify similarities.

We optimize the ranking log likelihood

$$L_{\text{rel}} = \sum_{\substack{(\mathbf{x},\mathbf{y}_g)\in D,\\ \mathbf{y}_r \sim D_{\mathbf{y}}}} \log \sigma(s_{\text{rel}}(\mathbf{x}, \mathbf{y}_g) - s_{\text{rel}}(\mathbf{x}, \mathbf{y}_r)), \tag{4.10}$$

where $\mathbf{y}_g$ is the gold ending and $\mathbf{y}_r$ is a randomly sampled ending.

Lexical Style Model

In practice RNNs generate text that exhibit much less lexical diversity than their training data. To counter this effect we introduce a simple discriminator based on observed lexical distributions which captures writing style as expressed through word choice. This classifier therefore encodes aspects of the maxim of Manner.

The scoring function is defined as

$$s_{\text{bow}}(\mathbf{y}) = \mathbf{w}_s^T \text{maxpool}(e(\mathbf{y})). \tag{4.11}$$

The model is trained with a ranking loss using negative examples sampled from the language model, similar to Equation 4.5.

	BookCorpus					TripAdvisor				
Model	BLEU	Meteor	Length	Vocab	Trigrams	BLEU	Meteor	Length	Vocab %	Trigrams
L2W	**0.52**	**6.8**	**43.6**	**73.8**	98.9	1.7	11.0	83.8	**64.1**	**96.2**
AdaptiveLM	**0.52**	6.3	43.5	59.0	92.7	**1.94**	**11.2**	**94.1**	52.6	92.5
CacheLM	0.33	4.6	37.9	31.0	44.9	1.36	7.2	52.1	39.2	57.0
Seq2Seq	0.32	4.0	36.7	23.0	33.7	1.84	8.0	59.2	33.9	57.0
SeqGAN	0.18	5.0	28.4	73.4	**99.3**	0.73	6.7	47.0	57.6	93.4
Reference	100.0	100.0	65.9	73.3	99.7	100.0	100.0	92.8	69.4	99.4

Table 4.1: Results for automatic evaluation metrics for all systems and domains, using the original continuation as the reference. The metrics are: Length - Average total length per example; Trigrams - % unique trigrams per example; Vocab - % unique words per example.

4.2.3 *Mixture Weight Learning*

Once all the communication models have been trained, we learn the combined decoding objective. In particular we learn the weight coefficients λ_k in equation 4.1 to linearly combine the scoring functions, using a discriminative loss

$$L_{\text{mix}} = \sum_{(\mathbf{x},\mathbf{y})\in D} (f_\lambda(\mathbf{x},\mathbf{y}) - f_\lambda(\mathbf{x},\mathcal{A}(\mathbf{x}))^2, \tag{4.12}$$

where $\mathcal{A}$ is the inference algorithm for beam search decoding. The weight coefficients are thus optimized to minimize the difference between the scores assigned to the gold continuation and the continuation predicted by the current model.

Mixture weights are learned online: Each successive generation is performed based on the current values of λ, and a step of gradient descent is then performed based on the prediction. This has the effect that the objective function changes dynamically during training: As the current samples from the model are used to update the mixture weights, it creates its own learning signal by applying the generative model discriminatively. The SGD learning rate is tuned separately for each dataset.

4.2.4 *Beam Search*

Due to the limitations of greedy decoding and the fact that our scoring functions do not decompose across time steps, we perform generation with a beam search procedure, shown in Algorithm 1. The naive approach would be to perform beam search based only on the language model, and then rescore the k best candidate completions with our full model. We found that this approach leads to limited diversity in the beam and therefore cannot exploit the strengths of the full model.

Instead we score the current hypotheses in the beam with the full decoding objective: First, each hypothesis is expanded by selecting the k highest scoring next words according to the language model (we use beam size $k = 10$). Then k sequences are sampled from the k^2 candidates according to the (softmax normalized) distribution over the candidate scores given by the full decoding objective. Sampling is performed in

BookCorpus	Specific Criteria				Overall Quality		
L2W vs.	Repetition	Contradiction	Relevance	Clarity	Better	Equal	Worse
AdaptiveLM	+0.48	+0.18	+0.12	+0.11	**47%**	20%	32%
CacheLM	+1.61	+0.37	+1.23	+1.21	**86%**	6%	8%
Seq2Seq	+1.01	+0.54	+0.83	+0.83	**72%**	7%	21%
SeqGAN	+0.20	+0.32	+0.61	+0.62	**63%**	20%	17%
LM vs. Reference	-0.10	-0.07	-0.18	-0.10	41%	7 %	**52%**
L2W vs. Reference	+0.49	+0.37	+0.46	+0.55	**53%**	18%	29%

TripAdvisor	Specific Criteria				Overall Quality		
L2W vs.	Repetition	Contradiction	Relevance	Clarity	Better	Equal	Worse
AdaptiveLM	+0.23	-0.02	+0.19	-0.03	**47%**	19%	34%
CacheLM	+1.25	+0.12	+0.94	+0.69	**77%**	9%	14%
Seq2Seq	+0.64	+0.04	+0.50	+0.41	**58%**	12%	30%
SeqGAN	+0.53	+0.01	+0.49	+0.06	**55%**	22%	22%
LM vs. Reference	-0.10	-0.04	-0.15	-0.06	38%	10%	**52%**
L2W vs. Reference	-0.49	-0.36	-0.47	-0.50	25%	18%	**57%**

Table 4.2: Results of crowd-sourced evaluation on different aspects of the generation quality as well as overall quality judgments. For each sub-criteria we report the average of comparative scores on a scale from -2 to 2. For the overall quality evaluation decisions are aggregated over 3 annotators per example.

order to increase diversity, using a temperature of 1.8, which was tuned by comparing the coherence of continuations on the validation set.

At each step, the discriminator scores are recomputed for all candidates, with the exception of the entailment score, which is only recomputed for hypotheses which end with a sentence terminating symbol. We terminate beam search when the *termination_score*, the maximum possible score achievable by terminating generation at the current position, is smaller than the current best score.

4.3 EXPERIMENTS

4.3.1 *Corpora*

We use two English corpora for evaluation. The first is the TripAdvisor corpus (Wang, Lu, and Zhai, 2010), a collection of hotel reviews with a total of 330 million words.[3] The second is the BookCorpus (Zhu et al., 2015), a 980 million word collection of novels by unpublished authors.[4] In order to train the discriminators, mixing weights, and the Seq2Seq and SeqGAN baselines, we segment both corpora into sections of length ten

3 http://times.cs.uiuc.edu/~wang296/Data/
4 http://yknzhu.wixsite.com/mbweb

sentences, and use the first 5 sentence as context and the second 5 as the continuation. See Appendix 4.9 for further details.

4.3.2 *Baselines*

ADAPTIVELM Our first baseline is the same Adaptive Softmax (Grave et al., 2016) language model used as base generator in our framework (§4.2.1). This enables us to evaluate the effect of our enhanced decoding objective directly. A 100k vocabulary is used and beam search with beam size of 5 is used at decoding time. AdaptiveLM achieves perplexity of 37.46 and 18.81 on BookCorpus and TripAdvisor respectively.

CACHELM As another LM baseline we include a continuous cache language model (Grave, Joulin, and Usunier, 2017) as implemented by Merity, Keskar, and Socher (2018), which recently obtained state-of-the-art perplexity on the Penn Treebank corpus (Marcus, Marcinkiewicz, and Santorini, 1993). Due to memory constraints, we use a vocabulary size of 50k for CacheLM. To generate, beam search decoding is used with a beam size 5. CacheLM obtains perplexities of 70.9 and 29.71 on BookCorpus and TripAdvisor respectively.

SEQ2SEQ As our evaluation can be framed as sequence-to-sequence transduction, we compare against a seq2seq model directly trained to predict 5 sentence continuations from 5 sentences of context, using the OpenNMT attention-based seq2seq implementation (Klein et al., 2017). Similarly to CacheLM, a 50k vocabulary was used and beam search decoding was performed with a beam size of 5.

SEQGAN Finally, as our use of discriminators is related to Generative Adversarial Networks (GANs), we use SeqGAN (Yu et al., 2017a) a GAN for discrete sequences trained with policy gradients.[5] This model is trained on 10 sentence sequences, which is significantly longer than previous experiments with GANs for text; the vocabulary is restricted to 25k words to make training tractable. Greedy sampling was found to outperform beam search. For implementation details see Appendix 4.8.

4.3.3 *Evaluation Setup*

We pose the evaluation of our model as the task of generating an appropriate continuation given an initial context. In our open-ended generation setting the continuation is not required to be a specific length, so we require our models and baselines to generate 5-sentence continuations, consistent with the way the discriminator and seq2seq baseline datasets are constructed.

Previous work has reported that automatic measures such as BLEU (Papineni et al., 2002) and Meteor (Denkowski and Lavie, 2010) do not lead to meaningful evaluation when used for long or creative text generation where there can be high variance among acceptable generation outputs (Vedantam, Lawrence Zitnick, and Parikh, 2015;

5 We use the implementation available at `https://github.com/nhynes/abc`.

Context	We thoroughly enjoyed staying at this hotel. We travelled with our kids aged 4 and 6. The rooms were clean, spacious and well maintained. The beds were great and all of us slept well. The hotel provided vouchers towards breakfast at the adjoining restaurant.
L2W	We didn't have breakfast, so can't comment on that. The concierge was very helpful, and gave us directions, etc. Overall, we would definitely recommend this hotel to others. We would stay here again in a heartbeat. If you're looking for a great location, and good value, I'd recommend this hotel.
AdaptiveLM	The breakfast buffet was very good, and the staff was very friendly and helpful, and it was a great way to start our day! I would definitely stay here again, and would recommend it to my friends, and would stay there again if I ever go back to Montreal. The location was great, and the staff was friendly and helpful. I would definitely stay here again!!
CacheLM	The staff was very friendly and helpful. The hotel is located in a great location. It is right in the middle of the gaslamp district. I would recommend this hotel to anyone. I would definitely stay here again.
Seq2Seq	The staff were very friendly and helpful. The hotel is in a great location. The staff were very friendly and helpful. The hotel is in a great location. The staff were very friendly and helpful.
SeqGAN	We had a breakfast at Shula's & a delicious breakfast. The staff was very helpful and helpful. The breakfast was great as well. The staff was very helpful and friendly. We had a great service and the food was excellent.
Reference	The restaurant was great and we used the vouchers towards whatever breakfast we ordered. The hotel had amazing grounds with a putting golf course that was fun for everyone. The pool was fantastic and we lucked out with great weather. We spent many hours in the pool, lounging, playing shuffleboard and snacking from the attached bar. The happy hour was great perk.

Table 4.3: Example continuations generated by our model (L2W) and various baselines (all given the same context from TripAdvisor) compared to the reference continuation.

Trip Advisor Ablation

Ablation vs. LM	Repetition	Contradiction	Relevance	Clarity	Better	Neither	Worse
REPETITION ONLY	+0.63	+0.30	+0.37	+0.42	**50%**	23%	27%
ENTAILMENT ONLY	+0.01	+0.02	+0.05	-0.10	39%	20%	**41%**
RELEVANCE ONLY	-0.19	+0.09	+0.10	+0.060	36%	22%	**42%**
LEXICAL STYLE ONLY	+0.11	+0.16	+0.20	+0.16	38%	25%	**38%**
ALL	+0.23	-0.02	+0.19	-0.03	**47%**	19%	34%

Table 4.4: Crowd-sourced ablation evaluation of generations on TripAdvisor. Each ablation uses only one discriminative communication model, and is compared to ADAPTIVELM.

Wiseman, Shieber, and Rush, 2017). However, we still report these measures as one component of our evaluation. Additionally we report a number of custom metrics which capture important properties of the generated text: *Length* – Average sequence length per example; *Trigrams* – percentage of unique trigrams per example; *Vocab* – percentage of unique words per example. Endings generated by our model and the baselines are compared against the reference endings in the original text. Results are given in Table 4.1.

For open-ended generation tasks such as our own, human evaluation has been found to be the only reliable measure (Li et al., 2016c; Wiseman, Shieber, and Rush, 2017). For human evaluation, two possible endings are presented to a human, who assesses the text according to several criteria, which are closely inspired by Grice's Maxims: repetition, contradiction, relevance and clarity. See Appendix 4.10 for examples of the evaluation forms we used. For each criterion, the two continuations are compared using a 5-point Likert scale, to which we assign numerical values of −2 to 2. The scale measures whether one generation is strongly or somewhat preferred above the other, or whether they are equal. Finally, the human is asked to make a judgement about overall quality: which ending is better, or are they of equal quality?

The human evaluation is performed on 100 examples selected from the test set of each corpus, for every pair of generators that are compared. We present the examples to workers on Amazon Mechanical Turk, using three annotators for each example. The results are given in Table 4.2. For the Likert scale, we report the average scores for each criterion, while for the overall quality judgement we simply aggregate votes across all examples.

4.4 RESULTS AND ANALYSIS

4.4.1 *Quantitative Results*

The absolute performance of all the evaluated systems on BLEU and Meteor is quite low (Table 4.1), as expected. However, in relative terms L2W is superior or competitive with all the baselines, of which ADAPTIVELM performs best. In terms of vocabulary

and trigram diversity only SeqGAN is competitive with L2W, likely due to the fact that sampling based decoding was used. For generation length only L2W and AdaptiveLM even approach human levels, with the former better on BookCorpus and the latter on TripAdvisor.

Under the crowd-sourced evaluation (Table 4.2), on BookCorpus our model is consistently favored over the baselines on all dimensions of comparison. In particular, our model tends to be much less repetitive, while being more clear and relevant than the baselines. AdaptiveLM is the most competitive baseline owing partially to the robustness of language models and to greater vocabulary coverage through the adaptive softmax. SeqGAN, while failing to achieve strong coherency, is surprisingly diverse, but tended to produce far shorter sentences than the other models. CacheLM has trouble dealing with the complex vocabulary of our domains without the support of either a hierarchical vocabulary structure (as in AdaptiveLM) or a structured training method (as with SeqGAN), leading to overall poor results. While the Seq2Seq model has low conditional perplexity, we found that in practice it is less able to leverage long-distance dependencies than the base language model, producing more generic output. This reflects our need for more complex evaluations for generation, as such models are rarely evaluated under metrics that inspect *characteristics* of the text, rather than ability to predict the gold or overlap with the gold.

For the TripAdvisor corpus, L2W is ranked higher than the baselines on overall quality, as well as on most individual metrics, with the exception that it fails to improve on contradiction and clarity over the AdaptiveLM (which is again the most competitive baseline). Our model's strongest improvements over the baselines are on repetition and relevance.

Ablation

To investigate the effect of individual discriminators on the overall performance, we report the results of ablations of our model in Table 4.4. For each ablation we include only one of the communication modules, and train a single mixture coefficient for combining that module and the language model. The diagonal of Table 4.4 contains only positive numbers, indicating that each discriminator does help with the purpose it was designed for. Interestingly, most discriminators help with most aspects of writing, but all except repetition fail to actually improve the overall quality over AdaptiveLM.

The repetition module gives the largest boost by far, consistent with the intuition that many of the deficiencies of RNN as a text generator lie in semantic repetition. The entailment module (which was intended to reduce contradiction) is the weakest, which we hypothesize is the combination of (a) mismatch between training and test data (since the entailment module was trained on SNLI and MultiNLI) and (b) the lack of smoothness in the entailment scorer, whose score could only be updated upon the completion of a sentence.

Crowd Sourcing

Surprisingly, L2W is even preferred over the *original* continuation of the initial text on BookCorpus. Qualitative analysis shows that L2W's continuation is often a straightforward continuation of the original text while the true continuation is more surprising and contains complex references to earlier parts of the book. While many of the issues of automatic metrics (Liu et al., 2016; Novikova et al., 2017) have been alleviated by crowd-sourcing, we found it difficult to incentivize crowd workers to spend significant time on any one datum, forcing them to rely on a shallower understanding of the text.

4.4.2 *Qualitative Analysis*

L2W generations are more topical and stylistically coherent with the context than the baselines. Table 4.3 shows that L2W, AdaptiveLM, and SeqGAN all start similarly, commenting on the breakfast buffet, as breakfast was mentioned in the last sentence of the context. The language model immediately offers generic compliments about the breakfast and staff, whereas L2W chooses a reasonable but less obvious path, stating that the previously mentioned vouchers were not used. In fact, L2W is the only system not to use the line *"The staff was very friendly and helpful."*, despite this sentence appearing in less than 1% of reviews. The semantics of this sentence, however, is expressed in many different surface forms in the training data (e.g., *"The staff were kind and quick to respond."*).

The CacheLM begins by generating the same over-used sentence and only produce short, generic sentences throughout. Seq2Seq simply repeats sentences that occur often in the training set, repeating one sentence three times and another twice. This indicates that the encoded context is essentially being ignored as the model fails to align the context and continuation.

The SeqGAN system is more detailed, e.g. mentioning a specific location "Shula's" as would be expected given its highly diverse vocabulary (as seen in Table 4.1). Yet it repeats itself in the first sentence. (e.g. *"had a breakfast", "and a delicious breakfast"*). Consequently SeqGAN quickly devolves into generic language, repeating the incredibly common sentence *"The staff was very helpful and friendly."*, similar to Seq2Seq.

The L2W models do not fix every degenerate characteristic of RNNs. The TripAdvisor L2W generation consists of meaningful but mostly disconnected sentences, whereas human text tends to build on previous sentences, as in the reference continuation. Furthermore, while L2W repeats itself less than any of our baselines, it still paraphrases itself, albeit more subtly: "we would definitely recommend this hotel to others." compared to "I'd recommend this hotel." This example also exposes a more fine-grained issue: L2W switches from using "we" to using "I" mid-generation. Such subtle distinctions are hard to capture during beam re-ranking and none of our models address the linguistic issues of this subtlety.

4.5 RELATED WORK

ALTERNATIVE DECODING OBJECTIVES A number of papers have proposed *alternative decoding objectives* for generation (Shao et al., 2017). Li et al. (2016) proposed a *diversity-promoting objective* that interpolates the conditional probability score with negative marginal or reverse conditional probabilities. Yu et al. (2017) also incorporate the reverse conditional probability through a noisy channel model in order to alleviate the *explaining-away* problem, but at the cost of significant decoding complexity, making it impractical for paragraph generation. Modified decoding objectives have long been a common practice in statistical machine translation (Chiang, Knight, and Wang, 2009; Koehn, Och, and Marcu, 2003; Och, 2003; Watanabe et al., 2007) and remain common with neural machine translation, even when an extremely large amount of data is available (Wu et al., 2016). Inspired by all the above approaches, our work presents a general learning framework together with a more comprehensive set of composite communication models.

PRAGMATIC COMMUNICATION MODELS Models for pragmatic reasoning about communicative goals such as Grice's maxims have been proposed in the context of referring expression generation (Frank and Goodman, 2012). Andreas and Klein (2016) proposed a neural model where candidate descriptions are sampled from a generatively trained *speaker*, which are then re-ranked by interpolating the score with that of the *listener*, a discriminator that predicts a distribution over choices given the speaker's description. Similar to our work the generator and discriminator scores are combined to select utterances which follow Grice's maxims. Yu et al. (2017) proposed a model where the speaker consists of a convolutional encoder and an LSTM decoder, trained with a ranking loss on negative samples in addition to optimizing log-likelihood.

GENERATIVE ADVERSARIAL NETWORKS GANs (Goodfellow et al., 2014) are another alternative to maximum likelihood estimation for generative models. However, backpropagating through discrete sequences and the inherent instability of the training objective (Che et al., 2017) both present significant challenges. While solutions have been proposed to make it possible to train GANs for language (Yu et al., 2017a) they have not yet been shown to produce high quality long-form text, as our results confirm.

GENERATION WITH LONG-TERM CONTEXT Several prior works studied paragraph generation using sequence-to-sequence models for image captions (Krause et al., 2017), product reviews (Dong et al., 2017; Lipton, Vikram, and McAuley, 2015), sport reports (Wiseman, Shieber, and Rush, 2017), and recipes (Kiddon, Zettlemoyer, and Choi, 2016). While these prior works focus on developing neural architectures for learning domain specific discourse patterns, our work proposes a general framework for learning a generator that is more powerful than maximum likelihood decoding from an RNN language model for an arbitrary target domain.

4.6 CONCLUSION

We proposed a unified learning framework for the generation of long, coherent texts, which overcomes some of the common limitations of RNNs as text generation models. Our framework learns a decoding objective suitable for generation through a learned combination of sub-models that capture linguistically-motivated qualities of good writing. Human evaluation shows that the quality of the text produced by our model exceeds that of competitive baselines by a large margin.

4.7 SUPPLEMENTARY A

4.7.1 *Base Language Model*

We use a 2-layer GRU (Cho et al., 2014) with a hidden size of 1024 for each layer. Following (Inan, Khosravi, and Socher, 2017) we tie the input and output embedding layers' parameters. We use an Adaptive Softmax for the final layer (Grave et al., 2016), which factorizes the prediction of a token into first predicting the probability of k (in our case $k = 3$) clusters of words that partition the vocabulary and then the probability of each word in a given cluster. To regularize we dropout (Srivastava et al., 2014) cells in the output layer of the first layer with probability 0.2. We use mini-batch stochastic gradient descent (SGD) and anneal the learning rate when the validation set performance fails to improve, checking every 1000 batches. Learning rate, annealing rate, and batch size were tuned on the validation set for each dataset. Gradients are backpropagated 35 time steps and clipped to a maximum value of 0.25.

4.7.2 *Cooperative Communication Models*

For all the models except the entailment model, training is performed with Adam (Kingma and Ba, 2015) with batch size 64 and learning rate 0.01. The classifier's hidden layer size is 300. Dropout is performed on both the input word embeddings and the non-linear hidden layer before classification with rate 0.5.

Word embeddings are kept fixed during training for the repetition model, but are fine-tuned for all the other models.

Entailment Model

We mostly follow the hyperparameters of Parikh et al. (2016): Word embeddings are projected to a hidden size of 200, which are used throughout the model. Optimization is performed with AdaGrad (Duchi, Hazan, and Singer, 2011) with initial learning rate 1.0 and batch size 16. Dropout is performed at rate 0.2 on the hidden layers of the 2-layer MLPs in the model.

Our entailment classifier obtains 82% accuracy on the SNLI validation set and 68% accuracy on the MultiNLI validation set.

Relevance Model

The convolutional layer is a one-dimensional convolution with filter size 3 and stride 1; the input sequences are padded such that the input and output lengths are equal.

4.8 SUPPLEMENTARY B

CACHELM Due to memory constraints, we use a vocabulary size of 50k for CACHELM. Beam search decoding is used, with a beam size 5.

SEQGAN The implementation we used adds a number of modelling extensions to the original SeqGAN. In order to make training tractable, the vocabulary is restricted to 25k words, the maximum sequence length is restricted to 250, Monte Carlo rollouts to length 4, and the discriminator updated once for every 10 generator training steps. Greedy decoding sampling with temperature 0.7 was found to work better than beam search.

SEQ2SEQ Due to memory constraints, we use a vocabulary size of 50k for SEQ2SEQ. Beam search decoding is used, with a beam size 5.

4.9 SUPPLEMENTARY C

For the language model and discriminators we use a vocabulary of 100, 000 words – we found empirically that larger vocabularies lead to better generation quality. To train our discriminators and evaluate our models, we use segments of length 10, using the first 5 sentences as context and the second 5 as the reference continuation. For TripAdvisor we use the first 10 sentences of reviews of length at least 10. For the BookCorpus we split books into segments of length 10. We select 20% of each corpus as held-out data (the rest is used for language model training). From the held-out data we select a test set of 2000 examples and two validation sets of 1000 examples each, one of which is used to train the mixture weights of the decoding objective. The rest of the held-out data is used to train the discriminative classifiers.

4.10 SUPPLEMENTARY D

The forms used on Amazon Mechanical Turk are pictured in Tables 4.5, 4.6, 4.7, and 4.8.

BookCorpus

Instructions

In this task you will be provided with 5 sentences from the middle of a book, and 2 possible continuations.

We'd like you to tell us whether you agree or disagree with the statements below, which compare and contrast the two continuations.

Thank you!

Task

Beginning Text: The cat figure leaned forward and spoke quietly his voice resonating in the room. You were in mortal danger when I chanced upon you in the hospital. The creature who was in that room was an assassin, sent by evil men to kill our queen. Jared looked puzzled and mordalayn paused. Do, do you mean sophie?

Continuation:

Continuation A What is wrong with you?" "I don't know. I didn't know. I don't know." "you have nothing to say.

Continuation B "I 'm sure he didn't mean to kill him." She shook her head. "no, it 's not. It was just a dream." "what happened?"

Question 1: Repetitiveness

Compare the **continuations** based on their **repetitiveness**. A good continuation **DOES NOT** repeat details from earlier in the continuation or from the first 5 sentences.

Repetition Good/Bad Examples (Expand/Collapse)

Beginning

"*...They'd been waiting for a while and Melissa had given in to temptation and was perusing a magazine rack in one of the mini-malls located a few feet from another institution of higher-thought removal; a fast-food chain designed to represent a slow-food eatery.*"

The following continuation is GOOD as it does not repeat previous content.

- "*The waiters and waitresses were all dressed to impress with the appropriate buttons and trinkets placed accordingly. The thought was to put the diner at ease by displaying humorous one-liners which would cause them to feel an attraction for the high-schoolers or employment-challenged and over-order in the process...*"

The following continuation is BAD as it repeats many words and phrases.

- "*The table was set with what looked to be a dining room, a dining table two chairs a table two chairs and a table The chairs were comfortable and comfortable...*"

Statement: Continuation A repeats itself MORE than Continuation B.

Strongly Agree | Agree | They're roughly the same | Actually, **Continuation B** repeats itself a little **MORE** | **Continuation B** repeats itself **much MORE** than **Continuation A**

Question 2: Contradiction

Compare the **continuations** based on how much they **contradict** themselves or the first 5 sentences. A good continuation should **NOT** contradict itself or the beginning 5 sentences.

Contradiction Good/Bad Examples (Expand/Collapse)

Beginning

"*...On a long stretch of narrow road, the truck stopped abruptly and all the men jumped out, disconnecting the trailer. The process took half a minute, then the truck did a three-point turn back toward the main route north. *"

The following continuation is GOOD as it does not contradict itself or the beginning sentences.

- "*The controllers at Hanscom almost lost contact before they realized the boat was gone. They could not tell that two men had also left the truck under the tree cover...*"

The following continuation is BAD as it contradicts itself as well as details described in the first 5 sentences.

- "*The trees were dense. They were in an open field.*"

Statement: Continuation A contradicts itself MORE than Continuation B.

Strongly Agree | Agree | They're roughly the same | Actually, **Continuation B** contradicts itself a little **MORE** | **Continuation B** contradicts itself **much MORE** than **Continuation A**

Table 4.5: The first half of the form for the BookCorpus human evaluation.

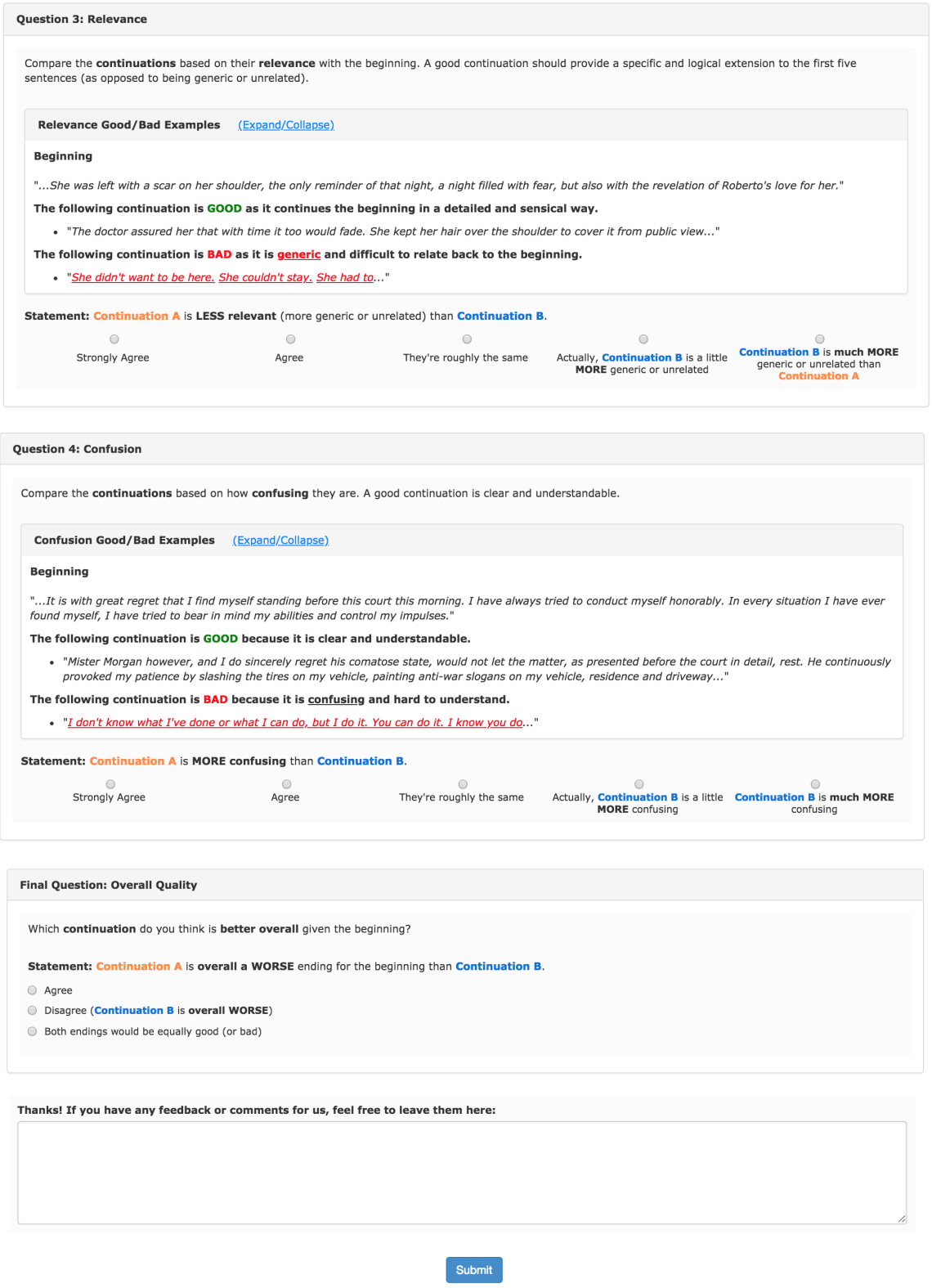

Table 4.6: The second half of the form for the BookCorpus human evaluation.

TripAdvisor

Instructions

In this task you will be provided with the first 5 sentences of a hotel review, and 2 possible continuations for the review.

We'd like you to tell us whether you agree or disagree with the statements below, which compare and contrast the two continuations.

Thank you!

Task

Beginning Text: We stayed at the hampton inn for a week in mid-aug and thought the hotel was very good. Had requested a room on a high floor that overlooked the pool, and this was exactly what we got. The room was clean and spacious, although surprisingly didn't have either a safe or a fridge. We had arrived in orlando in the middle of a heat wave with temperatures being in the mid 90s and really needed somewhere to cool our drinks, lotion, etc. So we ended up having to order a second ice bucket and use them as a makeshift drinks cooler!

Continuation:

Continuation A The staff at the front desk were very friendly, and helped us with our bags, even though we didn't use them, they were very helpful, and we had no problems. The hotel is in a great location, on the edge of the old town, and only about a 15 minute walk to the heart of the old city, and there is a bus stop right in front of the hotel, and it is just a short bus ride or a cab ride to the old town. All in all, we had a great time, and would definitely go back again, and would recommend this hotel to our friends, and wouldn't hesitate to stay here again. The only thing I would say is that you need to pay for wi-fi

Continuation B We also didn't drink the tap water (there's a tesco nearby! It's about $ 1. 50 to wash & restock, and they'll store your luggage if you're staying for more than one night and don't want to spend an arm and leg). The staff was very helpful in pointing out places to eat, etc. I would probably stay here again because of its location, but don't be fooled by the " free breakfast " - this hotel doesn't offer the " continental breakfast ", with " fresh " fruit juice, " cereal, " or " milk " and " do-it-yourself " waffles. I'm a sucker.

Question 1: Repetitiveness

Compare the **continuations** based on their **repetitiveness**. A good continuation **DOES NOT** repeat details from earlier in the continuation or from the beginning of the review.

Repetition Good/Bad Examples (Expand/Collapse)

Beginning

"*...The pool is very nice with a lazy river.*"

The following continuation is GOOD as it does not repeat previous content.

- "*Our kids had a good time. There's beach volleyball, bikes, golfing, etc....plus you get to enjoy the activities the Ritz has...*"

The following continuation is BAD as it repeats many words and phrases.

- "*The pool is great, and the gym is well equipped. The pool is very nice, and a nice place to relax...*"

Statement: Continuation A repeats itself MORE than Continuation B.

Strongly Agree	Agree	They're roughly the same	Actually, **Continuation B** repeats itself a little **MORE**	**Continuation B** repeats itself **much MORE** than **Continuation A**

Question 2: Contradiction

Compare the **continuations** based on how much they **contradict** themselves or the beginning of the review. A good continuation should **NOT** contradict itself or the beginning of the review.

Contradiction Good/Bad Examples (Expand/Collapse)

Beginning

"*...I did find the gentleman at the concierge desk a bit cold.*"

The following continuation is GOOD as it does not contradict itself or the beginning of the review.

- "*He told me to wait in line to print boarding passes but when a woman from the AAA convention being held there at the hotel came up he printed hers so she didn't have to wait in line...*"

The following continuation is BAD as it contradicts the beginning of the review.

- "*However, he was very friendly...*"

Statement: Continuation A contradicts itself MORE than Continuation B.

Strongly Agree	Agree	They're roughly the same	Actually, **Continuation B** contradicts itself a little **MORE**	**Continuation B** contradicts itself **much MORE** than **Continuation A**

Table 4.7: The first half of the form for the TripAdvisor human evaluation.

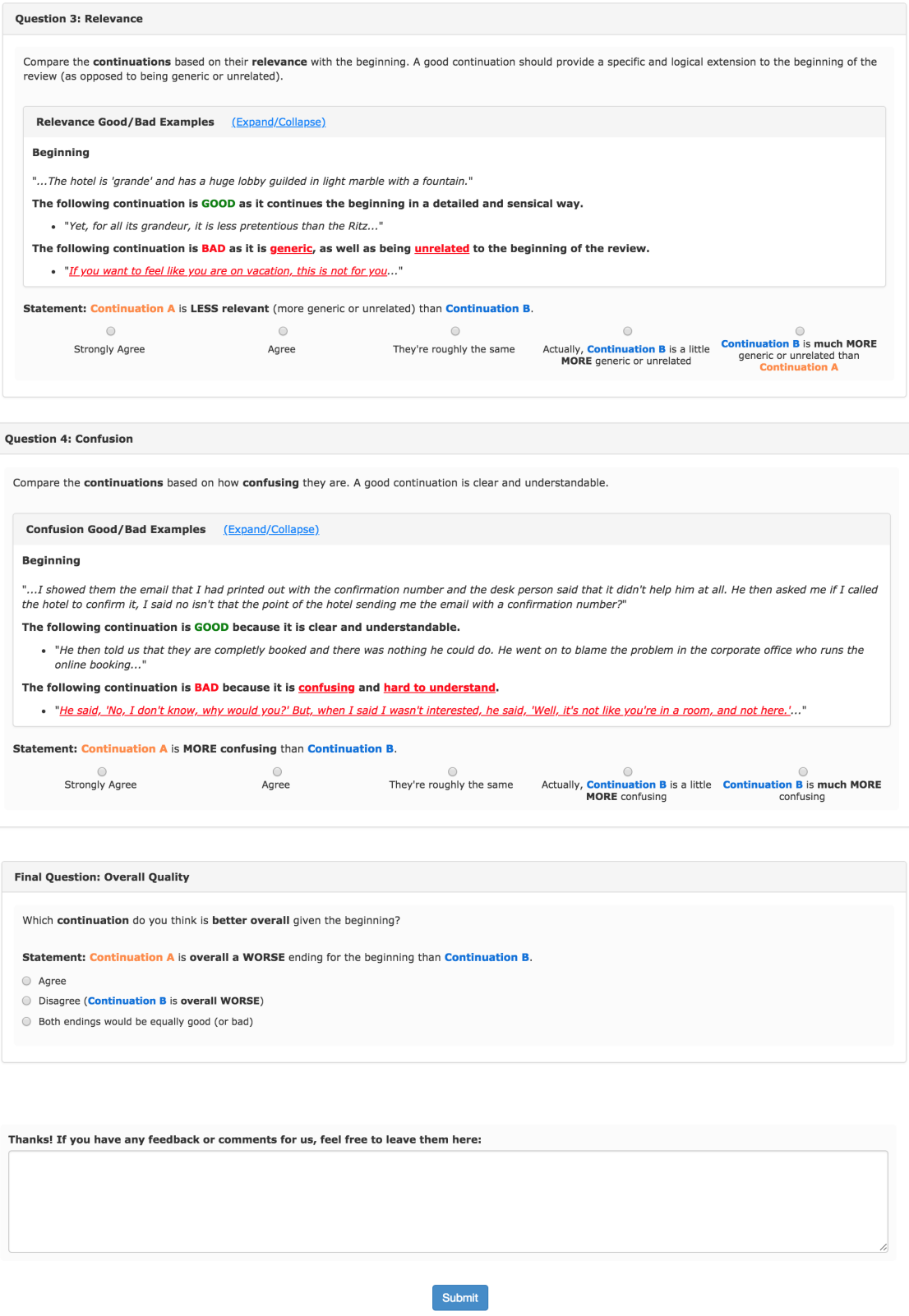

Table 4.8: The second half of the form for the TripAdvisor human evaluation.

5

GENERATIVE MODELS AS A COMPLEX SYSTEMS SCIENCE

This section is adapted from Holtzman, West, and Zettlemoyer (2023)

5.1 THE NEWFORMER: A THOUGHT EXPERIMENT

Consider the following thought experiment:

> Tomorrow, researchers at an industry lab publicly release a new kind of pretrained model: the Newformer. It has a completely different architecture than the Transformer (no attention, non-differentiable components, etc.), that outperforms all pretrained Transformers on the vast majority of benchmarks. Independent labs quickly verify that these results are sound, even on just-released benchmarks. While the composition of the training data is public, it is so expensive to train that no lab can afford to replicate it, even the one that produced it. Scaled-down versions do not exhibit the same performance or interesting behaviors as the original model. (A)

5.1.1 *How should we study the Newformer?*

Identifying high-level behaviors a model does or does not share with older models can steer us toward lower-level mechanisms it uses to solve tasks (§5.1.2, Figure 5.1). Interpretation techniques that rely on low-level details are model specific (§5.1.3) and often abandoned as the field changes. The Newformer is fictional, but it can help us reconceptualize the goals and methods of generative model research in light of the new landscape (§5.1.4).

How should we factorize model behavior into understandable and explanatory categories? (§5.2, Figure 5.2) We present a formalism for describing behavior (§5.2.1), noting that this corresponds to a *metamodel* that predicts aspects of a primary model (Figure 5.3). Benchmarks help us measure *performance*, but rarely *discover behavior* (§5.2.2) or *characterize* it (§5.2.3). Instead, discovered behaviors motivate new benchmarks (§5.2.4, Figure 5.4).

Generative models qualify as *complex systems* (§5.3), due to their *emergent behaviors* (§5.3.1, Figure 5.5), which are more often *discovered* than engineered (§5.3.2). A lack of clarity on *what* models do holds us back, as if we were studying organic chemistry without knowledge of biology (§5.3.3). This issue remains even when proprietary models are released (§5.3.4), as the problem lies in our lack of behavioral vocabulary; investigating possible mappings between training data and generated data can help us establish new behavioral categories (§5.3.5).

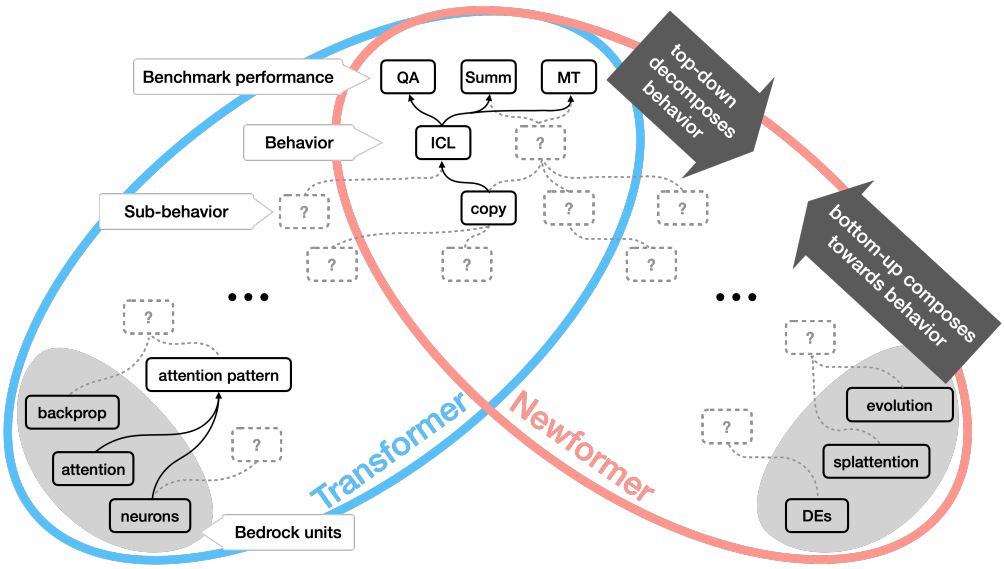

Figure 5.1: To explain why learned models self-organize the way they do from the bottom-up, it is useful to have top-down hierarchy of partially decomposed behaviors, to guide hypotheses with functionality we know the overall model has. While networks are composed of bedrock units for which we have a perfect understanding by construction (e.g. neurons), most emergent aspects of these systems are still undefined and undiscovered (represented as "?").

Despite the challenges, generative models are *easier* to study than many naturally arising complex systems (§5.4), because they are simulable by construction (§5.4.1). In contrast to physical phenomena, we can easily conduct a wide range of storable, repeatable experiments without observer effects (§5.4.2, Figure 5.7). We do, however, rely on the availability of open-source models (§5.4.3).

We conclude (§5.5) with an argument for increased focus on the foundational "what are models doing?" to guide the classic "why are models doing that?"

5.1.2 *Top-down behavioral taxonomy guides bottom-up mechanistic explanation*

The Newformer is a completely opaque result when considering benchmarks alone; it is simply better at doing what we want it to do than Transformer models (Vaswani et al., 2017) were before. However, as shown in Figure 5.1, a hierarchical taxonomy of LM behavior can guide our investigation of the Newformer, leading to questions such as:

1. What behaviors do the Transformer models and the Newformer model share, e.g., does the Newformer also repeat phrases more often than as seen in the training data?

2. Do they exhibit similar behaviors in the same contexts, e.g., does the Newformer need fewer input-output demonstrations to exhibit in-context learning at peak performance?

3. Do high-level behaviors decompose into the same lower-level behaviors or does the Newformer use different mechanisms to express them, e.g., when the Newformer is used for paraphrasing does it also tend to exactly copy the input?

Without such behavioral categories, we risk investigating the wrong direction when we try to interpret models, because we do not know what phenomena we are trying to explain in the first place.

Observed behavior can tell us where to look for bottom-up explanations. Al-Rfou et al. (2019) observed emergent copying behavior in Transformer Language Models (LMs), paving the way for the discovery of *copying heads* that make copying possible. Characterizing copying heads led to the discovery of *induction heads* (Elhage et al., 2021; Olsson et al., 2022): Transformer heads that are capable of copying abstract representational patterns in previous layers and appear to be responsible for in-context learning. Olsson et al. (2022) show that induction heads exhibit a variety of pattern matching behaviors that are still not fully catalogued.

Attempting to explain neural networks bottom-up without being guided by behavior can make it difficult to interpret results. For example, many works that identify anisotropy in the embedding spaces of large LMs diagnose this as a deficiency, and attempt to fix it (Ethayarajh, 2019; Gao et al., 2019; Wang et al., 2020). However, recent work suggests that this anisotropic property may not actually limit expressivity (Biś, Podkorytov, and Liu, 2021), may be a result of the transformer architecture specifically (Godey, Clergerie, and Sagot, 2023), and may actually be helpful for language models (Rudman and Eickhoff, 2023).

5.1.3 *The Transformer is the old Newformer*

A moderate Newformer event has occurred at least once before with LMs: the switch from Recurrent Neural Networks (RNNs) to Transformers. Despite many partial explanations (Hochreiter and Schmidhuber, 1997; Lakretz et al., 2019; Olah, 2015), we still lack an explanatory theory of how LSTM (Hochreiter and Schmidhuber, 1997) LMs such as ELMo (Peters et al., 2018) worked—what behaviors they could and could not capture, how these composed, etc.—even as they were replaced by models like BERT (Devlin et al., 2019) with similar use cases, but completely different architectural details. This does not bode well for the introduction of something like the Newformer which is significantly farther from the Transformer than the Transformer is from the LSTM.

On their own, bottom-up methods do not transfer well to new systems: analysis techniques that relied on mutated state and gating in RNNs, such as visualizing gating mechanisms (Karpathy, Johnson, and Fei-Fei, 2015), are not applicable to Transformers. Interpretation methods for Transformer models (Rogers, Kovaleva, and Rumshisky, 2020; Weiss, Goldberg, and Yahav, 2021), such as those that use attention, are unlikely to transfer over to the Newformer which breaks many previously immutable assumptions.

This suggests the value in doing more interpretation work that treats models like *black-box*es, as if we do not have access to their internal mechanisms. There is growing interest in looking at NLP systems as black-boxes (Bastings et al., 2022; Linzen

et al., 2019; Ribeiro et al., 2020), though much of this work still uses intermediate outputs—such as embeddings—rather than directly analyzing behavior in the output space models are trained to fit. Truly black-box methods can help insulate our analysis from change, giving us an anchor point that will always be testable on models that use the same modality (e.g., text, speech, images). Belinkov and Bisk (2017) show that neural machine translation systems are brittle to both natural spelling errors and synthetic character-level noise. This observation can be extended to ask: Is the Newformer robust to the same kinds of noise? Up to what threshold? Does the noise appear to be localized in the brittleness of tokenization, as was the case for Transformer-based systems (Provilkov, Emelianenko, and Voita, 2020)? Developing a rich inventory of such tests would give us a universal scaffolding for analyzing any Newformer the moment it is discovered.

5.1.4 *Are we there yet?*

Deciding whether a Newformer-like event has *truly* happened is an unresolvable question. New models are always partially derivative, and new (possibly artificial) axes can always be invented where they are worse or better (Wolpert and Macready, 1997). Yet three years on it is still infeasible for most labs to train a GPT-3 (Brown et al., 2020) level LM, costing approximately half a million dollars in compute alone for private companies—with engineering teams—to produce a similar model (Venigalla and Li, 2022). Thus it seems that the gap for training is only growing wider as ChatGPT (Schulman et al., 2022) and GPT-4 (OpenAI, 2023) become commonplace in research (Yang et al., 2023; Zhang et al., 2023), production (Eloundou et al., 2023; George and George, 2023; Ray, 2023), and even model evaluation (Liu et al., 2023b; Zheng et al., 2023).

Unlike the Newformer these models were never released, are frequently deprecated (OpenAI, 2023), change from day to day (Perry, 2023; Southern, 2023), and are know to be unstable over theoretically deterministic queries (Deng, 2023). Yet, the open source community has caught-up quickly (Alizadeh et al., 2023; Gunasekar et al., 2023; Mukherjee et al., 2023, *inter alia*) helped by industry labs' open-sourcing efforts (Almazrouei et al., 2023; Stability AI, 2023; Touvron et al., 2023, *inter alia*), and new finetuning tehchniques (Dettmers et al., 2023; Taori et al., 2023; Vicuna Team, n.d.).

However, the question still remains: how should we explain models not everyone can train? Models that are so arduous, slow, and expensive to train that we will likely never ablate all the necessary variables needed to study them properly?

This leaves us with mere behavior. We generally think of there as being two different kinds of behavior: the neural behavior of different activations in models and the "output" of the model in the form of human media (e.g., text, images, videos, etc.). Most methods of explaining models focus on the former: trying to explain why neural activations cluster into certain patterns and trying to understand what those patterns mean about the output.

We argue that not enough attention has been given to formalizing the latter: *what* models are doing in the first place, in terms of regularities in their outputs. Without such a formalization, bottom-up methods will have a much harder time deciding what precisely to explain, and what is simply noise.

5.2 THE BEHAVIORAL BOTTLENECK

How do we avoid proposing a new explanation for every exhibited difference? Surely we do not believe that we need a benchmark for every prompt that elicits slightly different behavior from a generative model? One solution is to propose many possible mechanisms, but make it an explicit research agenda to discover *the most parsimonious explanation*, a concept visualized in Figure 5.2. In other words, we want to be able to predict the aspects of text we care about (e.g. factuality) with the simplest rules possible. We briefly formalize this concept in §5.2.1, but the bulk of this chapter concerns the *need* for this new research focus and the perspective it yields.

Thousands of papers observe behavioral tendencies in models, such as the ability of a pretrained Transformer to copy from the input context (Al-Rfou et al., 2019; Elhage et al., 2021), which we will adopt as a running example. To understand models better, we must rigorously describe (1) what *aspect* of generative behavior a given mechanism predicts (e.g. repetition, copying from the training set, etc.) and (2) how much of the *information* in the output space of the model such predictions explain (since most will not predict 100% of what a model emits).

Figure 5.2 serves as a visual map of how we might explain models via behavior. On the top level we have a huge diversity of benchmarks that currently exist, and the even larger number that may one day exist. On the bottom we have the mathematical abstraction that describes the space of all possible models. Clearly both of these represent many more possibilities than is useful as an explanation or than is *necessary* to explain specific facets of model behavior. The intermediate levels, then, deal with simplified metamodels, i.e., models of the underlying generative model that are less explanatory, but still allow us to interpret or theorize around models.

5.2.1 *A working definition of "behavior"*

Fong and Vedaldi (2017) state that: "An explanation is a rule that predicts the response of a black box f to certain inputs." We think of a *behavior* as an explanation of limited aspects of a model, a concept we briefly formalize. We make reference to this formalization sparsely throughout the rest of the paper, as the argument can be understood without it, and we stress that the problem we are facing is more fundamental than a missing formalism.

Given a generative model from one input medium $\mathcal{X}$ (e.g., strings composed of at most 2048 tokens) and a source of randomness $\mathcal{R}$ to an output medium $\mathcal{Y}$ (e.g., 512x512 pixel images):

$$\mathcal{M} : \mathcal{X} \times \mathcal{R} \to \mathcal{Y} \tag{5.1}$$

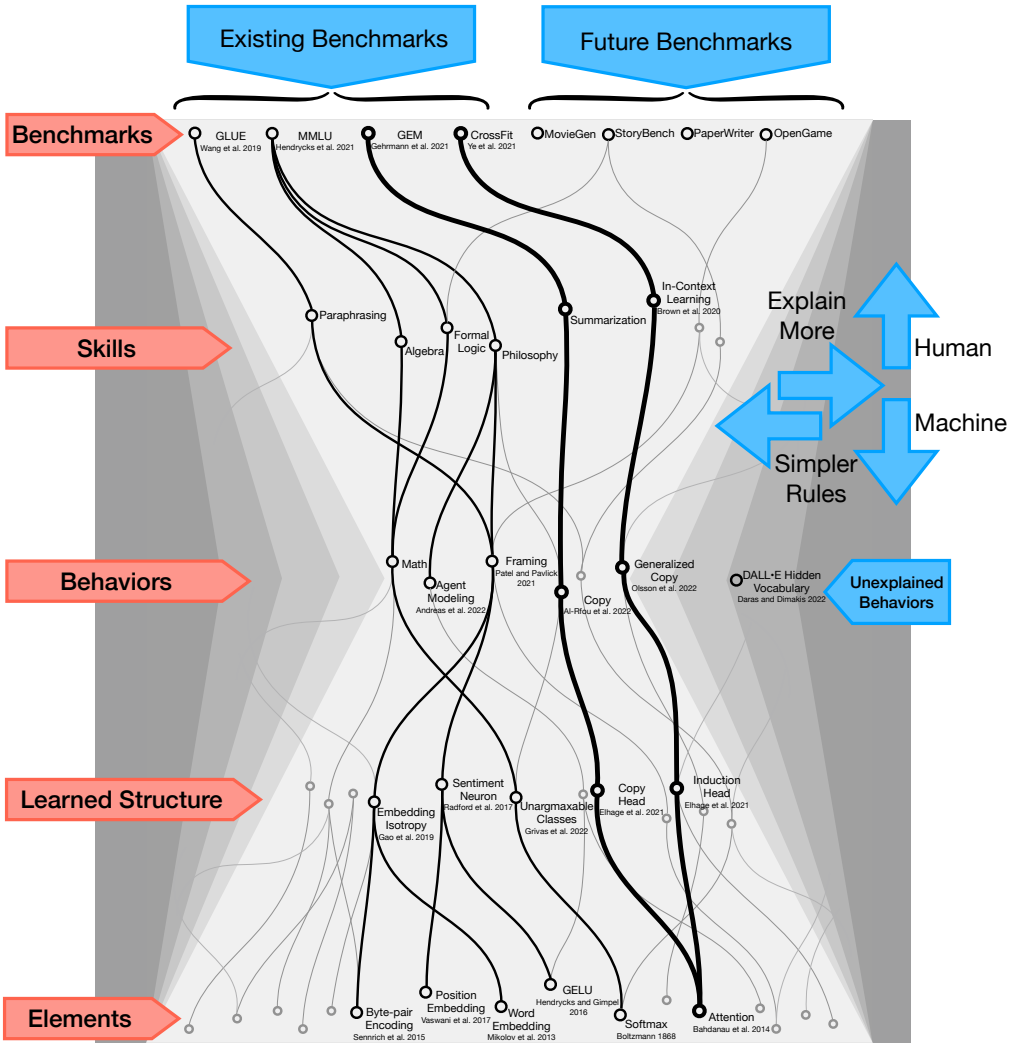

Figure 5.2: A visual representation of different aspects of models, shown from the basic elements of models on the bottom up to the benchmarks we are attempting to solve. Nodes represent invented and discovered aspects of models. The highlighted subgraph captures the concepts that we might want to use for understanding the phenomenon of "copying" in Transformers, when models generate sequences that appear in their local context window, a behavior that serves as a running example in this paper.

We might start out by noticing that Transformer models have higher scores on GEM (Gehrmann et al., 2021) (**Benchmark**), especially on summarization-like tasks (**Skill**). Inspecting the data generated by the models of interest, we might notice one of the qualitative differences separating Transformer models from other models is the ability to correctly use novel entities (Al-Rfou et al., 2019) (**Behavior**). We might ask why this is, embarking on an empirical study of when networks develop the ability to copy, as Elhage et al. (2021) did, discovering specific attention heads served as *copying heads* (**Learned Structure** supported by certain **Elements**). This led to other discoveries such as *induction heads* (Elhage et al., 2021; Olsson et al., 2022) (**Learned Structure**), which were found to perform a kind of *generalized copying* that supports inference-time pattern recognition (**Behavior**), e.g., for In-Context Learning (Brown et al., 2020) (**Skill**), leading to better results on fewshot benchmarks such as CrossFit (Ye, Lin, and Ren, 2021b) (**Benchmark**). Research can proceed by observing high-level behavioral regularities, explaining them via the tendencies of the model, and using this to achieve clarity about other observed behaviors.

we can define a behavior as a function from the same input medium to a feature set $\mathcal{F}$:

$$\mathcal{B} : \mathcal{X} \to \mathcal{F} \tag{5.2}$$

For instance, $\mathcal{M}$ may be a general purpose text-to-image model trained on scraped data, while $\mathcal{B}$ may map a string $x \in \mathcal{X}$ to a probability that an image $\mathcal{M}(x)$, contains at least one dog. Or $\mathcal{X}$ and $\mathcal{Y}$ may both be Unicode strings, in the case of an LM, with $\mathcal{B}$ being a binary prediction as to whether $\mathcal{M}(x)$ will eventually get caught in a repetition loop (Holtzman et al., 2020).

Our goal in proposing behaviors is to *explain* the underlying model using rules that capture model tendencies. Behaviors are explanatory to the extent that they give us information about the application of the model $\mathcal{M}$ under distributions $\mathcal{D}_{\mathcal{X}}$ and $\mathcal{D}_{\mathcal{R}}$ over $\mathcal{X}$ and $\mathcal{R}$, which we collectively refer to as $\mathcal{D}$ for brevity. We can formalize the notion of "giving us information about the application of the model" through the mutual information:

$$I_D(\mathcal{M}(\mathbf{X}, \mathbf{R}); \mathcal{B}(\mathbf{X})) = \tag{5.3}$$

$$H_D(\mathcal{M}(\mathbf{X}, \mathbf{R})) - H_D(\mathcal{M}(\mathbf{X}, \mathbf{R})|\mathcal{B}(\mathbf{X})) \tag{5.4}$$

where $\mathbf{X}$ and $\mathbf{R}$ are random variables drawn from $\mathcal{X}$ and $\mathcal{R}$ according to $\mathcal{D}_{\mathcal{X}}$ and $\mathcal{D}_{\mathcal{R}}$, and H is the entropy: $H(\mathbf{Y}) = \mathbb{E}_D[-\log p_D(\mathbf{Y})]$ for a random variable $\mathbf{Y}$. The mutual information is a direct measure of *how many bits of information we learn about one variable given another*, so this formulation directly tells us how much a behavior reveals about expected model output.

We call setting $\mathcal{D}_{\mathcal{X}}$ to be uniform over $\mathcal{X}$ the "mechanistic distribution." In this case, the mutual information is unrelated to the expected distribution of inputs in the wild, but is instead representative of how well we can model *any* input to $\mathcal{M}$. For instance, explaining an LM under the mechanistic distribution would require a behavior that predicts aspects of the LM's output accurately even for long strings of gibberish. This may be difficult, since we often use human linguistic features to make predictions about model outputs, but such behaviors are closer to the notion of mechanistic interpretability that tries to fully reverse engineer the model being studied (Olah, 2022).

If $\mathcal{M}$ ignores its source of randomness, i.e., $I(\mathcal{M}(\mathbf{X}, \mathbf{R}); \mathbf{R}) = 0$—as is the case for deterministic models such as a greedy-decoded LM—then the most explanatory behavior is simply $\mathcal{B} = \mathcal{M}(\mathbf{X}, r)$ for any $r \in \mathcal{R}$. This is a degenerate behavior, in that it is very explanatory, but has not brought us any closer to explaining $\mathcal{M}$. Therefore, we would like behaviors that are not just very high mutual information with the model, but also *point to predictable regularities* in $\mathcal{M}$, especially in a way that allows us to build up new hypotheses about it. Much has been written about what makes an explanation useful (Chen et al., 2023; Jacovi and Goldberg, 2020; Lipton, 2018, *inter alia*), and reviewing these desiderata is out of scope for this chapter.

The mutual information $I_D(\mathcal{M}(\mathbf{X}, \mathbf{R}); \mathcal{B}(\mathbf{X}))$ can also be viewed as *how much a behavior allows us to compress the output of a model* under a distribution $\mathcal{D}$, e.g., a distribution of articles for a summarization task (and a random number generator

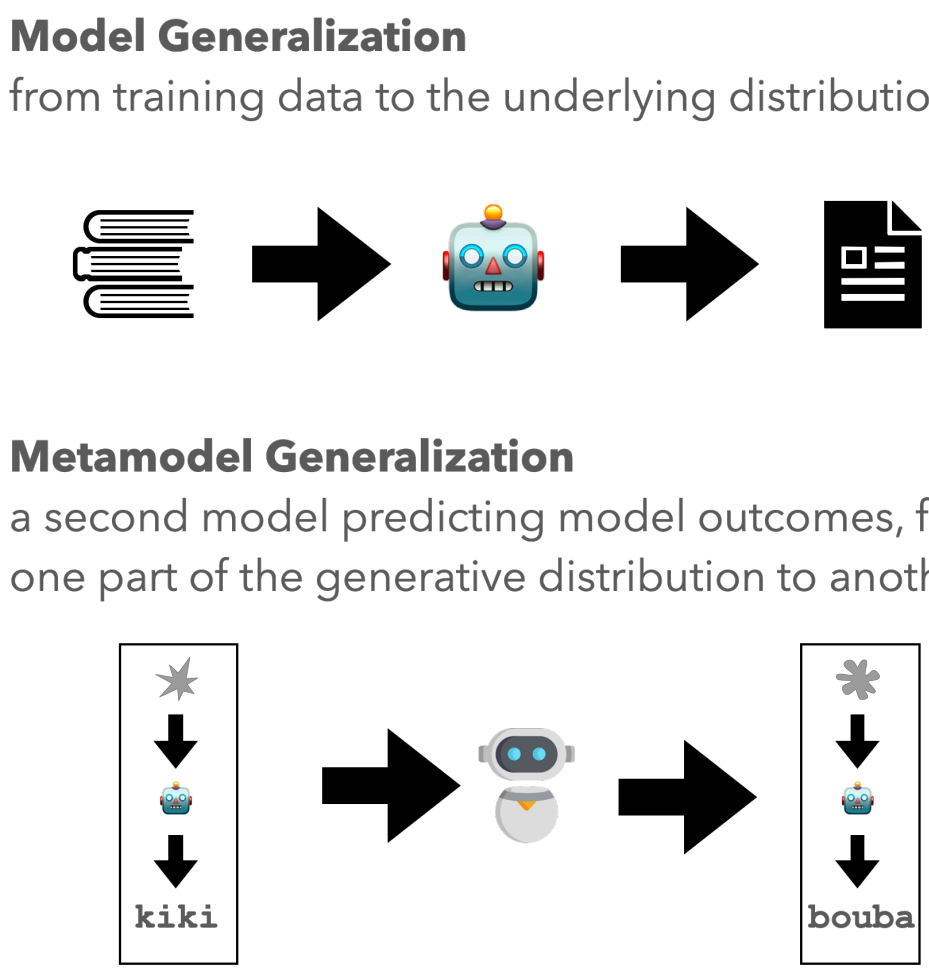

Figure 5.3: If we cannot train models easily, but those models are sufficiently general and useful, we can predict what models can and can't do, rather than what a model trained differently would do.

R). This is because any bits of information revealed by one variable, can be used to compress the other, under a proper coding scheme (Cover, 1999).

The concept of Minimum Description Length (MDL) has been used as an information theoretic criterion for finding good hypotheses (Grünwald, 2007). Essentially, it suggests an extension to Occam's razor (Barry, 2014): that we should favor explanations that are simple to describe and explain the object under study the most. We can formalize this notion for behaviors, via an encoding scheme C that represents behaviors $\mathcal{B}$ and outputs $y \in \mathcal{Y}$ as binary strings of variable (but finite) length $s \in \mathcal{S}$, where $|s|$ is the length of a string s. A naïve MDL objective would then be:

$$\underset{\mathcal{B}}{\operatorname{argmin}} |C(\mathcal{B})| + \sum_{x \in \mathcal{X}} \mathbb{E}_{D_{\mathcal{R}}} \left[|C(\mathcal{M}(x, \mathbf{R}) | \mathcal{B}(\mathbf{X}))| \right] \tag{5.5}$$

However, this would not suit our general objective: we do not necessarily wish to encode *all possible data a model could produce*, especially since most models have huge

output spaces of largely low probability density. Instead, we would like to quantify the information behaviors can save us under $\mathcal{D}_{\mathcal{X}}$.

To capture the idea "how much space does $\mathcal{B}$ save us under $\mathcal{D}_{\mathcal{X}}$ we can use:

$$\underset{\mathcal{B}}{\operatorname{argmin}} |\mathcal{C}(\mathcal{B})| + \alpha H_D(\mathcal{M}(\mathbf{X}, \mathbf{R})|\mathcal{B}(\mathbf{X})) \tag{5.6}$$

where we replace the second term with the conditional entropy H_D, since this describes the minimum number of bits that could be used to represent the information encoded (Cover, 1999). This can be interpreted as, "we would like behaviors that on average, save us more space in terms of encoding the possible outcomes of a model than they take to describe." α allows us to trade-off how much we weight the representation of the behavior vs. the outputs of a model, where larger values of α may be appropriate if we are dealing with a many outputs, making the bits saved by way of conditioning on the behavior more pertinent.

Overall, we seek to find behaviors that are both explanatory and simple to describe. We can think of this as attempting to find a *metamodel*: models that are designed or trained to predict another model's behavior (Barton and Meckesheimer, 2006), as illustrated in Figure 5.3. This suggests we want to find *behaviors that transfer over different contexts* so we can predict where models will be useful and where they will break down.

5.2.2 *Can benchmarks discover new behavior?*

In general, discrepancies in performance between benchmarks can *hint* at potentially new behavior, but they cannot discover behavior we have not yet observed. Given the diversity of NLP benchmarks, it is likely that the Newformer (§5.1) will perform drastically different on certain pairs of benchmarks we believe to be related, e.g., the same task in two different domains. This is a useful signal for where to inspect behavior, but benchmarks alone cannot reveal new abilities, underlying mechanisms, or shortcut heuristics the Newformer is relying on that *cause* a discrepancy in results and what else its effects are.

For example, it is very difficult to imagine how *prompting* (Liu et al., 2021; Radford et al., 2019c) could have been discovered via benchmarking. Finetuning a generative model, such as GPT-2 (a), and doing well or badly at any number of benchmarks could not have revealed a model can be prompted with text that matched training data patterns, to elicit behavior such as summary generation via the string "TL;DR" or translation through formatting such as "French sentence: <source>\\English sentence:". These discoveries are a result of inspecting the generative *behavior* of GPT-2, and only afterwards testing a perceived pattern on benchmarks.

How do we try to explain the behavior of models, once we know there's a discrepancy we want to explain? Often we attempt to look at the qualitative differences between tasks a model is good or bad at, and come up with hypotheses for what the model is failing to do when it performs poorly, e.g., across different finetuning tasks (Li et al., 2021). While useful for coming up with hypotheses, using benchmarks as evidence of behavior

requires care, because it is often unclear what a given benchmark is actually testing. Rohrbach et al. (2018) show that image captioning systems hallucinate objects not present in the scene, and are unintentionally *rewarded* for doing so by standard metrics, by capturing phrasing and *n*grams of reference captions better when hallucinating. Liao et al. (2021) describe a detailed framework for assessing benchmark validity and note the complexity of ensuring benchmarks test what we would like them to. Thus, since we often do not know precisely what behavior benchmarks test, they might indicate what contexts to examine the Newformer in, but not precisely what it does.

5.2.3 *Can benchmarks characterize behavior?*

Consider standardized tests for humans—such as the SAT (College Admission Counseling, 2008) or the NCEE (百度百科, 2022)—while the debate about how much these tests tell us is heated, there is little resistance to the statement: *test scores do not fully describe human behavior*, even within the subjects they test such as mathematics and biology.

Performance data about a bicycle is not sufficient to reverse-engineer its gear system. Even with perfectly valid benchmarks, the subspace of benchmark performance is not descriptive enough to characterize behavior. As we greatly increase the number of benchmarks, we are left with the problem of determining precisely how benchmarks overlap and differ in a way that characterizes behavior (Figure 5.2). Because the space of benchmarks is limited, as we test for human-desirable skills and human-interpretable pitfalls, discovering novel behavior in non-human systems is difficult.

Measuring systems only for their expected purposes makes it difficult to disentangle component behaviors that allow models to produce the desired or undesired outputs, as failure under distribution shift often reveals. For example, neural machine translation often outputs completely irrelevant translations under domain shift (Müller, Gonzales, and Sennrich, 2020; Wang and Sennrich, 2020). This is exacerbated by the fact that most generative models are not trained with a precise purpose in mind.

Imagine testing whether an LM can summarize an article. In order to summarize an article a requisite skill required by models is *copying*, because novel entities are constantly appearing, but need to be referenced in the summary. See, Liu, and Manning (2017) add a copying mechanism to an RNN in order to improve its copying ability for summarization. If we were to only look at performance on summarization, we would be unlikely to notice whether copying was happening or not directly—only whether performance is hitting certain desired levels.

Benchmarks are, by necessity, scoped to certain contexts that are presumed to test for certain behaviors—but they do not directly tell us what patterns the model is exploiting to solve the task, as Liao et al. (2021) point out. This was a hard-learned lesson in many benchmarks, such as when it was discovered that SNLI (Gururangan et al., 2018; Poliak et al., 2018) could be solved with *hypothesis-only* systems that only use a subset of the information that was supposed to be necessary to the task.

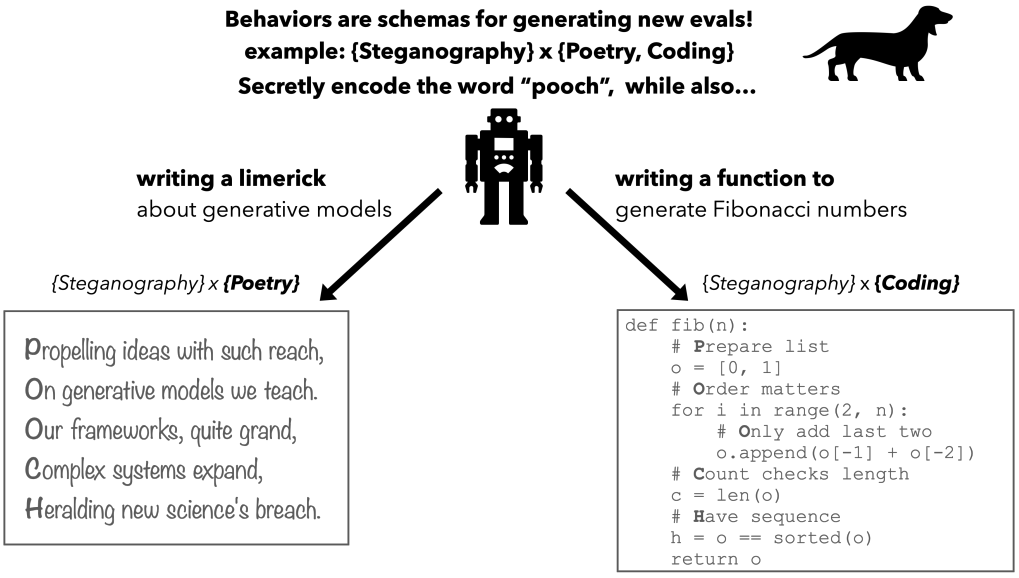

Figure 5.4: An example of how behaviors can be used to create new evaluations. These examples were generated from GPT-4, but required significant human curation, suggesting that Thought Experiment B has not yet occurred.

5.2.4 *Behaviors: building blocks for evaluation*

Benchmarks are still the best solution for coordinating cross-lab experimental *comparisons*, and we expect them to continue to be useful in that respect indefinitely. However, to answer "*What* strategy is the Newformer using for this task?" and "What failure modes should we expect?" and "What else do we expect the Newformer to be capable of?" we cannot use benchmarks alone to guide where we inspect model behavior, nor as a means to define it.

Instead, we propose an increased focus on behavior, because we believe the science of generative models is currently held back by insufficient understanding of *what* models are doing in general, rather than *how well* models perform on specific tasks. These are highly related to each other, and we can think of *behaviors as building blocks for evaluation*. Consider the following thought experiment:

> A new LM is released with many of the expected capabilities, such as basic arithmetic and basic translation, but another interesting behavior is noticed and hypothesized: when asked properly in natural language, the model can stegonographically encode complex hidden messages while completing other tasks. (B)

When this LM is released it is unlikely there are any benchmarks that test this particular capability. While we could design a specific benchmark for this behavior, this would be somewhat counter-productive: what we really care about is the *Cartesian product* of this behavior and other tasks that we were already testing. In this sense, behaviors are the building blocks for benchmarks.

As Chang and Bergen (2023) point out in their survey of behaviors, researchers are often surprised by the outputs of the models they work with; it should not surprise us that we cannot premeditate benchmarks to capture behavior when modeling improvements have outpaced our ability to be exposed to generated data. One way to be more nimble to new behaviors, is to directly measure behaviors we expect (Jain et al., 2023), flagging unexpected combinations for inspection.

On the surface, it might seem that naming behavioral categories such as "copying" or "in-context learning" is just as liable to obsolescence as any other analysis. What should we do if the Newformer does not exhibit these behaviors? We argue that this is a very unlikely scenario: as long as we are attempting to train models to mimic human understandable phenomena, there will be human perceivable patterns that we expect models to mimic as well.

5.3 GENERATIVE MODELS AS A COMPLEX SYSTEMS SCIENCE

While the Newformer (§5.1) is a thought experiment, it is representative of many facets of research regarding generative models today; suddenly, focus has shifted to searching for *emergent behaviors* in large and often inscrutable models. Larger pretrained models continue to be trained and continue to perform better (OpenAI, 2023; Pichai, 2023; Schulman et al., 2022, *inter alia*). While efforts to release models (Almazrouei et al., 2023; Stability AI, 2023; Touvron et al., 2023) and involve more researchers in model training (BigScience, 2022) can increase transparency and provide more information, it is well beyond the resources of the vast majority of labs to train. Efficiency breakthrough are likely to be exploited to further increase model size and feed into the same problem they were meant to solve.

Thus, it seems likely that training and re-training models is no longer the path towards understanding them for the vast majority of researchers. In many fields the creation of what it studies is impossible, from biology to astronomy. Many of these fields are *complex systems sciences*, in that they focus on the question illustrated in Figure 5.5: how do the macro-level behaviors we observe (life, black holes, etc.) arise from the micro-level units we understand better (chemicals, regular matter, etc.)?

In other words, we suggest studying *generative models themselves not just generative modeling*.

5.3.1 *What is a complex system?*

Newman (2011) establishes a working definition:

> [A] system composed of many interacting parts, such that the collective behavior of those parts together is more than the sum of their individual behaviors. The collective behaviors are..."emergent" behaviors, and a complex system can thus be said to be a system of interacting parts that displays emergent behavior.

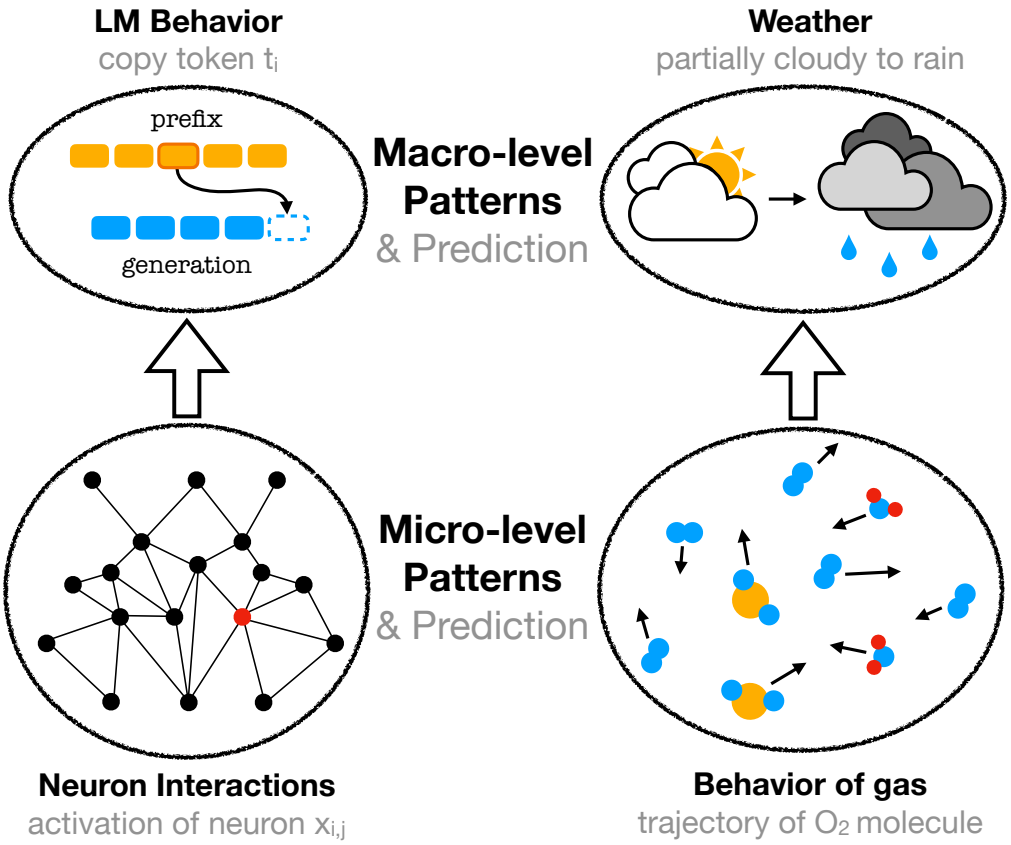

Figure 5.5: Complex systems are characterized by two or more levels of regularity: a micro-level in which local interactions are at least partially predictable and a macro-level in which many local interactions collectively exhibit recognizable patterns. *Emergence* describes how macro-level regularity is hard to predict in advance from comparatively well-understood micro-level dynamics.

Recently, interest in emergent behavior has grown in NLP (Bubeck et al., 2023; Manning et al., 2020; Teehan et al., 2022; Wei et al., 2022, *inter alia*), though it is usually defined, in terms of scaling over model parameters, dataset size, or computational power. We rely on a much simpler definition:

> Emergent behaviors are system level behaviors that are hard to predict from the dynamics of lower level subcomponents.

For instance, the ocean is a complex system. We can understand many properties of individual water molecules, e.g., H_2O has a partial positive and negative charge in certain places due its composition, but the aggregate properties of *water* as a collective whole exhibits predictable properties such as waves. It is difficult to predict the properties of water from H_2O because "the interactions of interest are non-linear...[yielding] levels of organization and hierarchies—selected aggregates at one level become 'building blocks' for emergent properties at a higher level, as when H_2O molecules become building blocks for water." (Holland, 2014)

Similarly, we understand the basic mechanical properties of LMs at the neuronal level, e.g., we have a perfect understanding of how to predict what any individual neuron will do given arbitrary inputs (by construction), but we also notice patterns at the level of *model behavior*, e.g., the emergent copying behavior, which is observed in both Transformer models (Al-Rfou et al., 2019; Khandelwal et al., 2019) and LSTM models (2018). In the face of new behavior that a model such as the Newformer might exhibit, we would be even less certain of how lower-level system components add up to observed responses.

5.3.2 *Emergent behaviors in LMs are discovered, not designed*

Neural architectural elements (e.g. position embeddings) and training methods (e.g. masking strategies) deeply affect the resulting model but do not fully explain behavior. We often fail to create the behavior we attempted to engineer into an architecture *and* discover new, unintentional behavior.

Many architectures have been designed to make use of longer context (Beltagy, Peters, and Cohan, 2020; Child et al., 2019; Yu et al., 2023, *inter alia*), but evidence suggests that these models often do not make use of the long-term dependencies that they intended to capture (Liu et al., 2023a; Press, Smith, and Lewis, 2021; Sun et al., 2021). Inversely, BERT was shown to capture much of the functionality of a knowledge base without task-specific training (Petroni et al., 2019).

To illustrate the difference between *designing* and *discovering* behavior, let us return to our running example of the copying behavior, where models produce a span that was in their input. A classic example of designing behavior is pointer-generator models (See, Liu, and Manning, 2017), in which a specific, discrete mechanism was added to encourage a certain behavior: copying. Transformers, on the other hand, were designed such that computation at a given time-step could *attend* to any previous time-step that was included in the context window. This intentionally removed the recurrence in architectures such as LSTMs (Hochreiter and Schmidhuber, 1997) and GRUs (Cho et al., 2014) in order to increase efficiency on highly parallelizable hardware such as GPUs and TPUs. A side-effect of this change was the emergent behavior of *copying* that arises directly from the Transformer architecture trained as a language model (Al-Rfou et al., 2019).

Instead of directly designing models for these purposes, we are now in the position of training general models with different structure and actively probing them for behavior. Using various data and masking strategies has produced models that can be controlled through different metadata (Aghajanyan et al., 2022; Keskar et al., 2019; Zellers et al., 2019b), while instruction-tuning has shown that pretrained LMs can be finetuned for control (Chung et al., 2022; Mishra et al., 2022; Ouyang et al., 2022, *inter alia*) often with very limited data (Dettmers et al., 2023; Taori et al., 2023; Zhou et al., 2023).

This *discovery* process focuses on giving the model access to certain kinds of correlations, and then inspecting what model behavior emerges.

5.3.3 *Neuronal explanations are limited by our understanding of behavior*

It is difficult to explain how or why LMs produce their outputs without having a good description of *what* they do. Explaining behavior bottom-up, requires an understanding of what behaviors we are trying to explain. Mittal, Diallo, and Tolk (2018) note:

> an emergent property of a system is usually discovered at the macro-level of the behavior of the system and cannot be immediately traced back to the specifications of the components, whose interplay produce this emergence.

This is the situation we find ourselves in with regards to large, pretrained models. We cannot, in general, predict how structure will form. While we can engineer systems with the hope of producing certain kinds of behavior, e.g., training on multimodal data to produce models that can draw inferences in ways that integrate paired text and images, this often does not produce the desired results (Ilharco et al., 2021; Parcalabescu et al., 2021).

Bottom-up investigation can reveal key properties of emergent organization within LMs, e.g. BERT replicates features of the classical NLP pipeline (Tenney, Das, and Pavlick, 2019). But when anomalous behavior is discovered, e.g., the DALL•E 2 hypothesized "hidden vocabulary" of invented words that correspond to specific image categories (Daras and Dimakis, 2022), it is difficult to investigate them with bottom-up tools until we reach a better understanding of what triggers them, what their scope is, etc. There have been attempts to reject the hidden vocabulary hypothesis (Hilton, 2022), but it is a very difficult hypothesis to rebut from first principles: what tests reject the hypothesis "DALL•E 2 has a hidden set of vocabulary with clear and consistent meaning" rather than "this specific mapping from the vocabulary to features isn't correct"?

This is similar to trying to research organic chemistry without knowledge of biology: it is certainly not impossible, but without high-level guides to the kind of structure one is expecting, the search space is huge and it is difficult to know where to look. Our lack of a behavioral taxonomy hampers research into internal structure, especially in models that break current assumptions such as the Newformer, as it is significantly more challenging to probe for structure without knowing what patterns in the outputs hint at the presence of structure.

5.3.4 *Access is not a silver bullet*

Consider the following thought experiment:

> Tomorrow, all industry labs publicly release all of their pretrained models (C)

Despite the fact that this would doubtlessly help us understand the basic properties of a given model such as ChatGPT, e.g., how large it is, we would still have significant obstacles on the way to explaining why ChatGPT is capable of writing short stories for almost any given prompt.

Indeed, the problem with answering the question of "How can a language model write a story?" has much less to do with language models and much more to do with

the fact that we are currently incapable of answering the question "How can x write a short story?" for any value of x. We find ourselves in the strange position of being able to train models we do not fully understand *for tasks we do not fully understand or anticipate in advance*.

The key to answering this question is to ask: what kind of explanation would satisfy us? For instance, when it comes to LMs, one explanation is that models are simply reconstructing long sequences from the training set and stitching them together. While a significant amount of memorization is taking place (Carlini et al., 2023; Lee et al., 2022; McCoy et al., 2023, *inter alia*) models appear to be able to generate data that is not a trivial recombination of the training data (Bubeck et al., 2023; Olsson et al., 2022; Tirumala et al., 2022).

The goal, then, should be to build up the case for a reasonable hypothesis that explain the breadth, depth, and (most importantly) mistakes models make when executing a complex task. However, we do not want a new explanation for every new task, which is precisely why we argue for the formalization and study of *behaviors* that describe the underlying strategies of models.

While model access would not directly solve these problems, we *do* believe that open-source models are a necessary prerequisite to this research program, for reasons outlined in §5.4.3.

5.3.5 *(Generated) data represents behavior*

Behavior in large pretrained models is nothing more than the answer to the question "How can we characterize the distribution of data this model generates?" Aspects of the training data such as the presence of multiple languages (Blevins and Zettlemoyer, 2022; Lin et al., 2021) or the number of repeated documents (Kandpal, Wallace, and Raffel, 2022; Lee et al., 2022) in the training set have been shown to be explanatory of zero-shot translation abilities and model tendency to leak training data, respectively.

Figure 5.6 visualizes what kind of behavioral mappings we can explore with data-based explanations. *Shared behavior*—patterns that are found in both the training and inference data (the outputs of the model)—are the simplest to search for, because they only require finding a specific behavior in the training or inference data and then looking for it in the other. For instance, the prompting behavior discovered in GPT-2 that causes summaries to be generated when "TL;DR" is placed after an article is an example of shared behavior. Idiosyncratic behaviors describe behaviors that don't appear to be caused directly by the training data at all, e.g., zero-length translations in many large models (Shi, Xiao, and Knight, 2020; Stahlberg and Byrne, 2019b; Stahlberg, Kulikov, and Kumar, 2022). Perhaps the most difficult to find behavioral mappings are those for which behavior in the corpus yields different behavior in the model, *contingent behavior*, as is hypothesized to be the case for DALL-E 2's "hidden vocabulary": nonsense words that appear to consistently lend certain meanings to produced images (Daras and Dimakis, 2022). Finally, unexpressed behavior is observed in the training data, but not in the inference data, such as long-term consistency in

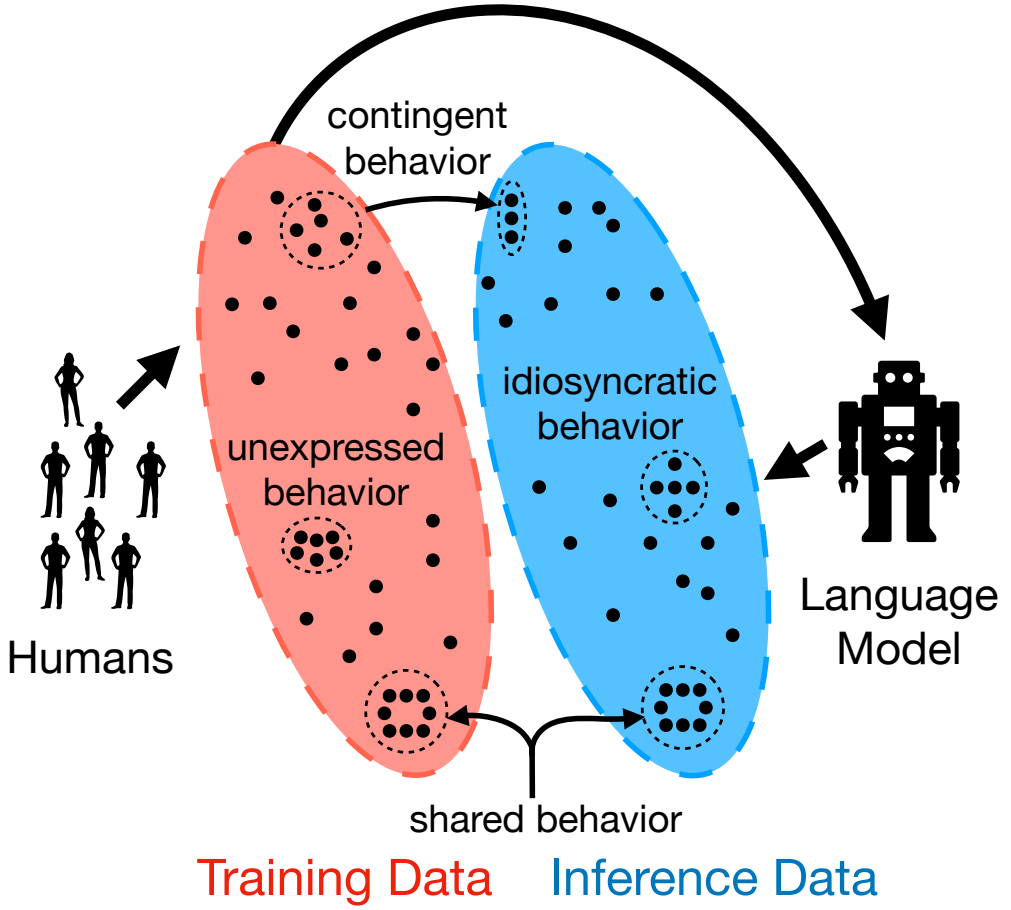

Figure 5.6: Generative language models are trained to capture the distribution of training data, then exhibit behavior in model outputs, i.e., *inference data*. See §5.3.5 for examples of the different behavioral mappings.

story telling (See et al., 2019; Xie, Cohn, and Lau, 2023) that models have yet to properly mimic for very long documents.

5.4 A DIFFERENT KIND OF COMPLEX SYSTEM

One reasonable worry is that taking on the complex systems lens will be fruitless because studying complex systems is a very difficult task, and we are not equipped to tackle such a hard problem.

In fact, compared to other complex systems, such as the brain, understanding current generative models is an immensely *easier* challenge, and can help us develop tools for the future. Turning our attention to "What, precisely, do language models do?" over "What is the best recipe for training large models?" we can take full advantage of the *complete simulability* of generative models. In the long run, it seems it will become more difficult to address the latter question coherently without better answers to the former.

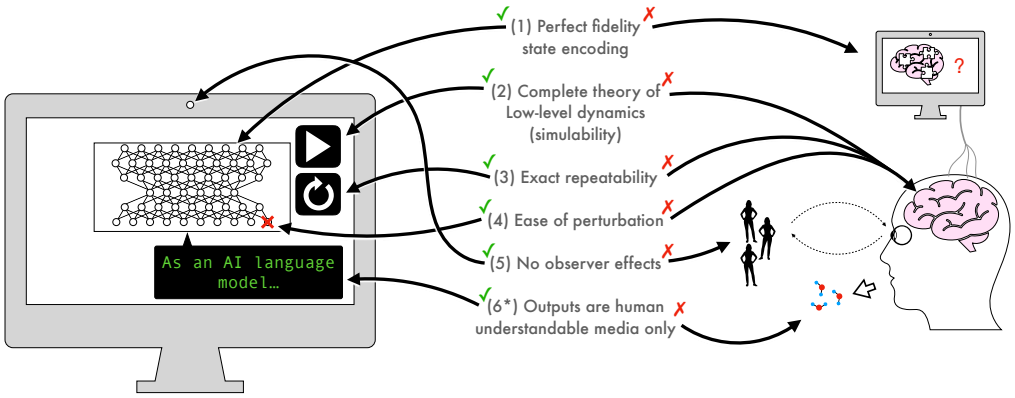

Figure 5.7: Visual representation of the advantages generative model researchers have over researchers that study the main other media generating system on earth: human beings.

5.4.1 *Two kinds of complex systems simulations*

> Complex systems theory is divided between two basic approaches. The first involves the creation and study of simplified mathematical models that, while they may not mimic the behavior of real systems exactly, try to abstract the most important qualitative elements into a solvable framework from which we can gain scientific insight...The second approach is to create more comprehensive and realistic models, usually in the form of computer simulations, which represent the interacting parts of a complex system, often down to minute details, and then to watch and measure the emergent behaviors that appear. (Newman, 2011)

At first glance, generative models can largely be described as the second complex systems approach: we train models to capture properties of the natural distribution of human media, such as internet text, images, videos, etc., and then attempt to study the emergent effects. Yet, this would be a mischaracterization of what, e.g., language models do: we do not expect that language models learn language the way a human does nor create languages the way the human species did.[1]

Instead, the triumphs of generative models are the result of emergent behavior within computational models trained to predict very general objectives. Many have been surprised that by learning on massively more data from a given medium than a human is ever exposed to, generative models can learn uncannily human patterns from simple, passive word prediction, denoising objectives, etc.

Generative models are certainly a computational simulation, but they are a simulation of an entire medium rather than a singular process we have isolated. We suggest thinking of generative models as a different kind of complex system, where *underlying patterns of a medium are learned by a model through optimization, and we then look for*

1 Though this certainly describes facets of certain subfields such as emergent communication (Lazaridou and Baroni, 2020), with recent work taking advantage of pretrained models (Steinert-Threlkeld et al., 2022), and developmentally plausible pretraining such as the recent BabyLM challenge (Warstadt et al., 2023).

those patterns within the model. Below we list ways in which this discovery process is made easier because our system of interest is the computational model itself, rather than a naturally arising system.

5.4.2 *Generative Models: the easiest complex system to study*

(1) **Perfect fidelity state encoding** Because neural networks are formal mathematical models, represented by code and parameters, there is zero *necessary ambiguity* in our representations. Imperfect data archiving and complex code bases often make it difficult to perfectly recover the formal model, but with sufficient effort, it is possible to store every bit of information about the state of the model at every computation step. We cannot track every neuron in a participant's brain at every moment due to the limited nature of our measurement instruments, but we can perfectly record the state of an LM in order to look for and verify emergent behavior, without influencing the system as we note in Advantage (5).

(2) **Complete theory of low-level dynamics** While Advantage (1) establishes that we can perfectly store the state of a generative model at any given moment in time, Advantage (2) notes that we have a perfect, mechanistic, and deterministic understanding of how one state of a model evolves into the next, unlike in physical experiments. Artificial neural networks do not need to be simulated, they are defined in a medium for simulation: executable code. Unlike in physics, where centuries of research have been spent chasing the bottom of the chain of causation, we *begin* with the base-level causal structure of the model. This does not follow directly from (1). It is possible to imagine a scenario where every static state is recordable, but where the rules that govern the changes in states are hard to discover, e.g., the problem of learning video game dynamics from pixels (Hafner et al., 2019). In practice, nondeterminism exists in certain fast computations (Morin and Willetts, 2020), but this can be removed at the cost of speed.

(3) **Exact repeatability** Directly entailed by (1) and (2) is the fact that experiments can be repeated *exactly*. An algorithm that uses randomness to generate text, may generate different text on a second run, but as long as the probabilities of different tokens are recorded the likelihood of that text (and of alternative branching paths) can be verified to be exactly the same. A psychologist who conducts a study twice will almost never get results that are exactly the same, simply because sample differences and unmeasured variables have to be accounted for. We distinguish repeatability from the broader notion of *replicability*, which also includes replicating a study to the level of detail described by the authors, leaving room for both human and systematic error. With proper code, data, and model releases, many generative model experiments are exactly repeatable, allowing us to reach for a much higher standard for replicability.

(4) **Ease of perturbation** We also have a complete description of *all possible models* given a certain setup, e.g., all possible combinations of weights for a given architecture. Combining this with Advantage (2), we can perturb a model of interest, and play out experiments with this new model *without destroying the original model.* Contrast this with studying human language production, for which most perturbations of the human brain are both unethical and illegal, partially because humans cannot be unperturbed. This allows for extremely targeted experiments, e.g., finding which weights in a network control a certain decision boundary.

(5) **No observer effects**—a classic problem in many complex systems is that by attempting to make a measurement one changes the value being measured, e.g., Clever Hans a horse that could allegedly play chess, but was simply reading the audience's reaction to possible moves (Prinz, 2006). In contrast, generative models do not distinguish between the same input given for different reasons or with different expectations by the experimenter. The caveat is that experimenters still control the input distributions to experiments allowing for systematic bias that accidentally leaks *experimenter expectations* (Rosenthal, 1976) to the model, as past research has consistently shown (Gururangan et al., 2018; McCoy, Pavlick, and Linzen, 2019; Poliak et al., 2018, *inter alia*). We must be careful about "tells" (Caro, 2003): stylistic and semantic artifacts that make it into the data which can give the model information the experimenter assume it does not have access to. Yet, the guarantee that the specific observer will not change the result is strong.

Advantages (1) and (2) allow us to completely remove any worry about hidden variables that may explain effects we attempted to explain through other means. (3) and (4), allow us to experiment freely, knowing that experiments and models that have been properly recorded are recoverable, leaving us free to perturb and explore the local neighborhood of similar models and setups. (5) partially relieves us of the fear of influencing the outcome through our means of observation, a key issue in many experiments involving language.

Another advantage, that does not apply to every generative model, deserves an honorable mention:

(6*) **(Some) generative models exclusively output human understandable media** Many complex systems, such as cities and brains, produce human understandable media as some percentage of their output. Many generative models produce human understandable media as their *only* output, an enormous advantage for two reasons. First, humans are better suited to positing patterns in human understandable media than, say, subatomic particles. Second, *the uncanny valley effect* (Mori, 2012) allows us to see when patterns are "almost correct but not quite" much more easily in human-related artifacts. While we sometimes finetune models to produce outputs that are no longer human understandable, by and large current generative models operate entirely within human media—and

we believe there is much that can be learned from this that will transfer over to generative models of other media.

Advantage (6*) is very special. By allowing us to take advantage of our intuitive understanding of media, it becomes easier to seek out the ways generated media diverges from the natural human media we are steeped in from infancy. Indeed, most named behaviors are failure modes, e.g. degeneration (Holtzman et al., 2020) or empty translations (Stahlberg and Byrne, 2019b)).

Organic chemistry has given a great deal to biology, but is very much indebted to it as well. Our hope is that we can take inspiration from these other complex system sciences to start taking the problem of understanding *behavior* seriously, as a distinct abstraction that needs to be decomposed and theorized, while putting our enormous advantages to good use.

5.4.3 *The necessity of open-source models*

Most of these advantages rely on stable access to a consistent representation of a model, which is difficult to guarantee via a proprietary API.

~~(1)~~ **Perfect fidelity state encoding** It is difficult to work with or guarantee saved state is persistent and untampered without direct access to said state. Even cryptographically signed state can be tampered *after* re-submission to an API for use there, making guarantees moot.

~~(2)~~ **Complete theory of low-level dynamics** With only imperfect knowledge of an underlying model, researchers must make assumptions about low-level dynamics in a model that may only partially be true, or possibly even completely false.

~~(3)~~ **Exact repeatability** In practice, it is impossible to guarantee that an API will not drift over time, something observed with even the apparent attempts at stable APIs in recent years (Deng, 2023).

~~(4)~~ **Ease of perturbation** It is normally impossible to perturb a model through an API, though some APIs allow for finetuning and special versions of models. However, the real issue is that it is impossible to ensure that such perturbations do precisely what they are claiming to do to the model, without access to the model or even the model architecture in most cases.

~~(5)~~ **No observer effects** Sadly, even though this is one of the greatest advantages of generative models, it is the one most destroyed by using models via APIs: companies consistently, and often silently, fix undesired (from the company's perspective) behaviors in models (Eliaçık, 2023; Kiho, 2023; Wilson, 2023) so that testing a certain hypothesis tends to influence future tests.

~~(6*)~~ **(Some) generative models exclusively output human understandable media** Without complete access to a model it is impossible to know if it doesn't have other outputs (or inputs) that would help explain the model's behavior more fully.

In short, without access to open-source models, these advantages are largely moot. However, the community has seen a consistent open-source releases of better generative models in many different media (Le et al., 2023; Luo et al., 2023; Rombach et al., 2021). There is unquestionably lag in the capabilities between proprietary and open-source models, and this is out of necessity: open-source cannot outpace private industry when private industry controls most of the training resources and can build on top of anything open-source does. But the fact that open source often lags only a year or two behind in terms of capabilities, and the fact that private labs are often incentivized to open-source models as a recruiting and market strategy, suggests that open-source will continue to be a wellspring of fascinating generative models to study. Indeed, if all progress stopped now, we believe it would be decades before we finished cataloging all of the generalizable behavioral principles with the hundreds of large generative models that have already been released; perhaps our successes would encourage future open-source releases.

5.5 CONCLUSION

How should we study models of data, when we don't fully understand the models or the data? We should study them first by asking *what* models do, before attempting the more complicated *how* and the bottomless question of *why*?

In this chapter, we presented a thought experiment: the Newformer, a model that would be impossible to study with many of the techniques we use to understand Transformer models today.

We argue that focusing on what *behaviors* explain its performance across tasks will lead-us to a deeper understanding of generative models' tendencies and guide bottom-up mechanistic explanation, as well as forming building blocks for evaluations.

We discuss how generative models are well captured by the definition of a complex system, due to the emergent behaviors they exhibit. This separates generative models from traditional machine learning, where models often served as explanations via behaviors that were architected directly into them. This opens up the need for *metamodels* that help us predict regularities in generative model outputs in order to understand them better.

While the prospect of studying models we do not have a clear understanding of is daunting, we highlight advantages that generative models have over naturally arising complex systems. These advantages, however, require open-source models as a prerequisite, a point we emphasize as a necessity for conducting replicable science.

5.6 LIMITATIONS

We present one perspective on the kind of science NLP is becoming, and how we can leverage the complex systems lens in order to better explore the phenomena we find ourselves faced with: generative models we do not fully understand. We cite evidence from NLP publications, blog posts, and other media, but this necessarily does not capture the totality of perspectives.

Indeed, we purposefully avoid attempting any sort of survey of these issues, as this would involve citing thousands of papers and be a very unwieldy object. Instead, we attempt to form an argument as economically as possible, attempting to put forth a new set of goals and principles for how to study generative models given current progress.

We make comparisons with other sciences and cite sources from those sciences where appropriate, but are extremely limited in expressing many equally relevant connections and in fully exploring the connections we do mention. There is an enormous amount related to sister fields (e.g., cognitive science, linguistics, etc.), other sciences that study complex systems (e.g., chemistry, biology, etc.), and regarding more meta-science issues (e.g., complex systems theory, chaos theory, etc.) that we could not cover, and we do not in any way attempt to—giving a complete account of these connections is simply beyond the reach of any one work.

Finally, parts of our assessment is necessarily subjective. We attempt to lay out the evidence as we see it, tracing the connections we drew in order to describe a style of research that we believe is necessary to face the current challenges of our field. This seems especially pertinent in a time when most researchers cannot train large generative models from scratch, but are excited to contribute to their study. With evidence drawn from the literature, we describe the current research space as we perceive it, and our vision for where it might go. Our hope is that this will add to a discussion on what the study of generative models currently is and what we, as a community, would like it to become.

6

CONCLUSION

In this dissertation we presented three projects that revealed aspects of current generative models of language that are often misinterpreted, and showed new techniques to better align models with the intended objective. We ended by considering the future of the overall study of generative models of human media, considering how we should think about this emerging field going forward.

Generative models are changing on the pace of weeks and the need for analysis tools that will continue to be useful as the field changes is growing just as fast. The success of complex models we do not fully understand has shifted NLP towards studying *emergent behaviors* in these models, which is definitional of *complex systems sciences*. Emergent organization in neural network weights must explain these behaviors, but it is difficult to know what structure to look for without a *taxonomy* of behaviors that they result in.

In this dissertation we have focused on revealing a few of these behaviors, drawing out their origins in the interaction of model components, and showing how we can make use of what models already do once we have a better understanding of model behavior. However, what we still lack is a systematic way of discovering and categorizing such behavior.

This leaves us with the question: what tools should we use to decompose language model behavior, if not benchmarks that measure performance on human-desired tasks? Language models exhibit many behaviors which are either detrimental or completely orthogonal to the tasks we as users and practitioner wish to complete. How can we discover these and what should our hypothesis set for such behaviors look like? In other words, what kinds of patterns can help us explain language model behavior, instead of anthropomorphizing models as if they were humans?

Traditionally, analysis in NLP has tended to focus on error categories that are either a predictable result of the model architecture or a human-inspired error category for the task at hand that we might expect a person to make. However, models and model architectures are no longer predictable enough for this to be sufficiently comprehensive, and so many new tasks are being proposed that cross-task error categories are necessary. We have presented an initial framework for thinking about these questions in Chapter 5, but the work of putting it into practice very much remains.

The patterns underlying the error cases and idiosyncrasies of generative models of language remain largely opaque. Why do models succeed on some tasks and fail on other, seemingly simpler ones? Why can a slight shift in input format or prompt result in completely different outputs? These deviations from human behavior pose a mystery, but also an opportunity to concretely discover the true mechanisms working within the models of today. Perhaps more importantly, they give us an opportunity to find where our specification for the tasks themselves are insufficient for us to make these distinctions.

Now that we have seen what generative models are capable of, and how differently they work than what we expected, a new field must rise to answer the question: what, precisely, do generative models do?

BIBLIOGRAPHY

Ackley, David H, Geoffrey E Hinton, and Terrence J Sejnowski (1985). "A learning algorithm for Boltzmann machines." In: *Cognitive science* 9.1, pp. 147–169.

Aghajanyan, Armen, Dmytro Okhonko, Mike Lewis, Mandar Joshi, Hu Xu, Gargi Ghosh, and Luke Zettlemoyer (2022). "HTLM: Hyper-Text Pre-Training and Prompting of Language Models." In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=P-pPW1nxf1r.

Al-Rfou, Rami, Dokook Choe, Noah Constant, Mandy Guo, and Llion Jones (2019). "Character-level language modeling with deeper self-attention." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33, pp. 3159–3166.

Alizadeh, Meysam, Maël Kubli, Zeynab Samei, Shirin Dehghani, Juan Diego Bermeo, Maria Korobeynikova, and Fabrizio Gilardi (July 2023). "Open-Source Large Language Models Outperform Crowd Workers and Approach ChatGPT in Text-Annotation Tasks." In: arXiv: 2307.02179 [cs.CL].

Almazrouei, Ebtesam, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Merouane Debbah, Etienne Goffinet, Daniel Heslow, Julien Launay, Quentin Malartic, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo (2023). "Falcon-40B: an open large language model with state-of-the-art performance." In.

Andreas, Jacob (2022). "Language Models as Agent Models." In: *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 5769–5779.

Andreas, Jacob and Dan Klein (2016). "Reasoning about Pragmatics with Neural Listeners and Speakers." In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, pp. 1173–1182. DOI: 10.18653/v1/D16-1125. URL: http://www.aclweb.org/anthology/D16-1125.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (2015a). "Neural Machine Translation by Jointly Learning to Align and Translate." In: *Proceedings of the 2015 International Conference on Learning Representations*.

– (2015b). "Neural Machine Translation by Jointly Learning to Align and Translate." In: *International Conference on Learning Representations*.

– (2015c). "Neural machine translation by jointly learning to align and translate." In.

Barry, C M (May 2014). *Who sharpened Occam's Razor?* en. https://www.irishphilosophy.com/2014/05/27/who-sharpened-occams-razor/. Accessed: 2023-7-7.

Barton, Russell R and Martin Meckesheimer (2006). "Metamodel-based simulation optimization." In: *Handbooks in operations research and management science* 13, pp. 535–574.

Bastings, Jasmijn, Yonatan Belinkov, Yanai Elazar, Dieuwke Hupkes, Naomi Saphra, and Sarah Wiegreffe (2022). "Proceedings of the Fifth BlackboxNLP Workshop on

Analyzing and Interpreting Neural Networks for NLP." In: *Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*.

Belinkov, Yonatan and Yonatan Bisk (2017). "Synthetic and Natural Noise Both Break Neural Machine Translation." In: *International Conference on Learning Representations*.

Beltagy, Iz, Matthew E Peters, and Arman Cohan (2020). "Longformer: The long-document transformer." In: *arXiv preprint arXiv:2004.05150*.

Bender, Emily M, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell (2021). "On the dangers of stochastic parrots: Can language models be too big?" In: *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pp. 610–623.

Bengio, Yoshua, Réjean Ducharme, Pascal Vincent, and Christian Janvin (2003). "A Neural Probabilistic Language Model." In: *Journal of Machine Learning Research* 3, pp. 1137–1155.

BigBench, Collaboration (2021). "Beyond the imitation game: Measuring and extrapolating the capabilities of language models." In: *In preparation*. URL: https://github.com/google/BIG-bench/.

BigScience (2022). "BigScience Model Training Launched." In: *BigScience Blog*.

Biś, Daniel, Maksim Podkorytov, and Xiuwen Liu (2021). "Too much in common: Shifting of embeddings in transformer language models and its implications." In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5117–5130.

Blevins, Terra and Luke Zettlemoyer (2022). "Language Contamination Explains the Cross-lingual Capabilities of English Pretrained Models." In: *arXiv preprint arXiv:2204.08110*.

Boltzmann, Ludwig (1868). "Studien über das Gleichgewicht der lebendigen Kraft zwischen bewegten materiellen Punkten [Studies on the balance of living force between moving material points]." In: *Wiener Berichte* 58, pp. 517–560.

Bommasani, Rishi, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. (2021). "On the opportunities and risks of foundation models." In: *arXiv preprint arXiv:2108.07258*.

Bowman, Samuel R., Gabor Angeli, Christopher Potts, and Christopher D. Manning (2015). "A Large Annotated Corpus for Learning Natural Language Inference." In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, pp. 632–642. DOI: 10.18653/v1/D15-1075. URL: http://www.aclweb.org/anthology/D15-1075.

Brown, Tom B, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. (2020). "Language models are few-shot learners." In: *arXiv preprint arXiv:2005.14165*.

Bubeck, Sébastien, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. (2023). "Sparks of artificial general intelligence: Early experiments with gpt-4." In: *arXiv preprint arXiv:2303.12712*.

Caccia, Massimo, Lucas Caccia, William Fedus, Hugo Larochelle, Joelle Pineau, and Laurent Charlin (2018). "Language GANs Falling Short." In: *Critiquing and Correcting*

Trends in Machine Learning: NeurIPS 2018 Workshop. URL: http://arxiv.org/abs/1811.02549.

Cai, Zheng, Lifu Tu, and Kevin Gimpel (2017). "Pay attention to the ending: Strong neural baselines for the roc story cloze task." In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 616–622.

Carlini, Nicholas, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang (2023). "Quantifying memorization across neural language models." In: *The Twelfth International Conference on Learning Representations*.

Caro, Mike (2003). *Caro's book of poker tells*. Cardoza Publishing.

Chang, Tyler A and Benjamin K Bergen (2023). "Language model behavior: A comprehensive survey." In: *arXiv preprint arXiv:2303.11504*.

Che, Tong, Yanran Li, Ruixiang Zhang, R. Devon Hjelm, Wenjie Li, Yangqiu Song, and Yoshua Bengio (2017). "Maximum-Likelihood Augmented Discrete Generative Adversarial Networks." In: *CoRR* abs/1702.07983. arXiv: 1702.07983. URL: http://arxiv.org/abs/1702.07983.

Chen, Chacha, Shi Feng, Amit Sharma, and Chenhao Tan (2023). "Machine Explanations and Human Understanding." In: *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*, pp. 1–1.

Chen, Yining, Sorcha Gilroy, Andreas Maletti, Jonathan May, and Kevin Knight (June 2018). "Recurrent Neural Networks as Weighted Language Recognizers." In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 2261–2271. DOI: 10.18653/v1/N18-1205. URL: https://www.aclweb.org/anthology/N18-1205.

Chiang, David, Kevin Knight, and Wei Wang (2009). "11,001 New Features for Statistical Machine Translation." In: *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Boulder, Colorado: Association for Computational Linguistics, pp. 218–226. ISBN: 978-1-932432-41-1. URL: http://dl.acm.org/citation.cfm?id=1620754.1620786.

Child, Rewon, Scott Gray, Alec Radford, and Ilya Sutskever (2019). "Generating long sequences with sparse transformers." In: *arXiv preprint arXiv:1904.10509*.

Cho, Kyunghyun, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio (2014). "On the Properties of Neural Machine Translation: Encoder–Decoder Approaches." In: *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pp. 103–111.

Chopra, Sumit, Michael Auli, and Alexander M. Rush (2016). "Abstractive Sentence Summarization with Attentive Recurrent Neural Networks." In: *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, pp. 93–98. URL: http://www.aclweb.org/anthology/N16-1012.

Chung, Hyung Won, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. (2022). "Scaling instruction-finetuned language models." In: *arXiv preprint arXiv:2210.11416*.

Clark, Christopher, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova (2019). "BoolQ: Exploring the Surprising Difficulty of Natural Yes/No Questions." In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2924–2936.

Clark, Elizabeth, Yangfeng Ji, and Noah A. Smith (June 2018). "Neural Text Generation in Stories Using Entity Representations as Context." In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 2250–2260. DOI: 10.18653/v1/N18-1204. URL: https://www.aclweb.org/anthology/N18-1204.

Clark, Peter, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord (2018). "Think you have solved question answering? try arc, the ai2 reasoning challenge." In: *arXiv preprint arXiv:1803.05457*.

College Admission Counseling, National Association for (2008). *Report of the commission on the use of standardized tests in undergraduate admission*. ERIC Clearinghouse.

Cover, Thomas M (1999). *Elements of information theory*. John Wiley & Sons.

Dagan, Ido, Oren Glickman, and Bernardo Magnini (2005). "The pascal recognising textual entailment challenge." In: *Machine Learning Challenges Workshop*. Springer, pp. 177–190.

– (2006). "The PASCAL Recognising Textual Entailment Challenge." In: *Proceedings of the First International Conference on Machine Learning Challenges: Evaluating Predictive Uncertainty Visual Object Classification, and Recognizing Textual Entailment*. MLCW'05. Berlin, Heidelberg: Springer-Verlag, pp. 177–190. ISBN: 3-540-33427-0, 978-3-540-33427-9. DOI: 10.1007/11736790_9. URL: http://dx.doi.org/10.1007/11736790_9D.

Daras, Giannis and Alexandros G Dimakis (2022). "Discovering the Hidden Vocabulary of DALLE-2." In: *arXiv preprint arXiv:2206.00169*.

Dauphin, Yann N, Angela Fan, Michael Auli, and David Grangier (2017). "Language modeling with gated convolutional networks." In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, pp. 933–941.

Davison, Joe, Joshua Feldman, and Alexander Rush (Nov. 2019). "Commonsense Knowledge Mining from Pretrained Models." In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 1173–1178. DOI: 10.18653/v1/D19-1109. URL: https://www.aclweb.org/anthology/D19-1109.

De Marneffe, Marie-Catherine, Mandy Simons, and Judith Tonhauser (2019). "The CommitmentBank: Investigating projection in naturally occurring discourse." In: *proceedings of Sinn und Bedeutung*. Vol. 23, pp. 107–124.

Deng, Yuntian (Mar. 2023). *OpenAI Watch*. en. https://openaiwatch.com/. Accessed: 2023-7-6, source: https://twitter.com/yuntiandeng/status/1641108596510343168?s=20.

Denkowski, Michael and Alon Lavie (2010). "Extending the METEOR Machine Translation Evaluation Metric to the Phrase Level." In: *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 250–253. URL: https://www.aclweb.org/anthology/D12-1097.

Dettmers, Tim, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer (2023). "Qlora: Efficient finetuning of quantized llms." In: *arXiv preprint arXiv:2305.14314*.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186.

Dong, Li, Shaohan Huang, Furu Wei, Mirella Lapata, Ming Zhou, and Ke Xu (Apr. 2017). "Learning to Generate Product Reviews from Attributes." In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. Valencia, Spain: Association for Computational Linguistics, pp. 623–632. URL: http://www.aclweb.org/anthology/E17-1059.

Duchi, John, Elad Hazan, and Yoram Singer (2011). "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization." In: *Journal of Machine Learning Research* 12.Jul, pp. 2121–2159.

Elhage, Nelson, Neel Nanda, Catherine Olsson, Tom Henighan†, Nicholas Joseph†, Ben Mann†, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah‡ (2021). "A Mathematical Framework for Transformer Circuits." In: *Transformer Circuits Thread*. URL: https://transformer-circuits.pub/2021/framework/index.html.

Eliaçık, Eray (2023). *Playing with fire: The leaked plugin DAN unchains ChatGPT from its moral and ethical restrictions*. https://dataconomy.com/2023/03/31/chatgpt-dan-prompt-how-to-jailbreak-chatgpt/. Accessed: 2023-7-6.

Eloundou, Tyna, Sam Manning, Pamela Mishkin, and Daniel Rock (2023). "Gpts are gpts: An early look at the labor market impact potential of large language models." In: *arXiv preprint arXiv:2303.10130*.

Erk, Katrin, Sebastian Padó, and Ulrike Padó (2010). "A Flexible, Corpus-Driven Model of Regular and Inverse Selectional Preferences." In: *Computational Linguistics* 36.4, pp. 723–763. DOI: 10.1162/coli_a_00017. URL: https://www.aclweb.org/anthology/J10-4007.

Ethayarajh, Kawin (2019). "How Contextual are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings." In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the*

9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 55–65.

Fan, Angela, Mike Lewis, and Yann Dauphin (2018). "Hierarchical Neural Story Generation." In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vol. 1, pp. 889–898.

Ficler, Jessica and Yoav Goldberg (2017). "Controlling Linguistic Style Aspects in Neural Language Generation." In: *Proceedings of the Workshop on Stylistic Variation*, pp. 94–104.

Fong, Ruth C and Andrea Vedaldi (2017). "Interpretable explanations of black boxes by meaningful perturbation." In: *Proceedings of the IEEE international conference on computer vision*, pp. 3429–3437.

Frank, Michael C. and Noah D. Goodman (2012). "Predicting Pragmatic Reasoning in Language Games." In: *Science* 336.6084, pp. 998–998. DOI: 10.1126/science.1218633. URL: http://science.sciencemag.org/content/336/6084/998.

Gao, Jun, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tieyan Liu (2019). "Representation Degeneration Problem in Training Natural Language Generation Models." In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=SkEYojRqtm.

Gao, Tianyu, Adam Fisch, and Danqi Chen (2021). "Making Pre-trained Language Models Better Few-shot Learners." In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 3816–3830.

Gehrmann, Sebastian, Tosin Adewumi, Karmanya Aggarwal, Pawan Sasanka Ammanamanchi, Anuoluwapo Aremu, Antoine Bosselut, Khyathi Raghavi Chandu, Miruna-Adriana Clinciu, Dipanjan Das, Kaustubh Dhole, et al. (2021). "The GEM Benchmark: Natural Language Generation, its Evaluation and Metrics." In: *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, pp. 96–120.

George, A Shaji and AS Hovan George (2023). "A Review of ChatGPT AI's Impact on Several Business Sectors." In: *Partners Universal International Innovation Journal* 1.1, pp. 9–23.

Godey, Nathan, Éric de la Clergerie, and Benoît Sagot (2023). "Is Anisotropy Inherent to Transformers?" In: *arXiv preprint arXiv:2306.07656*.

Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio (2014). "Generative Adversarial Nets." In: *Advances in Neural Information Processing Systems*, pp. 2672–2680.

Gordon, Jonathan and Benjamin Van Durme (2013). "Reporting bias and knowledge acquisition." In: *Proceedings of the 2013 workshop on Automated knowledge base construction*. ACM, pp. 25–30.

Goyal, Tanya, Junyi Jessy Li, and Greg Durrett (2022). "News Summarization and Evaluation in the Era of GPT-3." In: *arXiv preprint*.

Grave, Edouard, Armand Joulin, Moustapha Cissé, David Grangier, and Hervé Jégou (2016). "Efficient Softmax Approximation for GPUs." In: *arXiv preprint arXiv:1609.04309*.

Grave, Edouard, Armand Joulin, and Nicolas Usunier (2017). "Improving Neural Language Models with a Continuous Cache." In: *International Conference on Learning Representations*. URL: http://arxiv.org/abs/1612.04426.

Grice, H Paul (1975). "Logic and Conversation." In: *Speech Acts*. Ed. by P Cole and J L Morgan. Vol. 3. Syntax and Semantics. New York: Academic Press, pp. 41–58.

Grice, H Paul, Peter Cole, Jerry Morgan, et al. (1975). "Logic and Conversation." In: *1975*, pp. 41–58.

Grivas, Andreas, Nikolay Bogoychev, and Adam Lopez (May 2022). "Low-Rank Softmax Can Have Unargmaxable Classes in Theory but Rarely in Practice." In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, pp. 6738–6758. DOI: 10.18653/v1/2022.acl-long.465. URL: https://aclanthology.org/2022.acl-long.465.

Grünwald, Peter D (2007). *The minimum description length principle*. MIT press.

Guadarrama, Sergio, Niveda Krishnamoorthy, Girish Malkarnenkar, Subhashini Venugopalan, Raymond Mooney, Trevor Darrell, and Kate Saenko (2013). "Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition." In: *Proceedings of the IEEE international conference on computer vision*, pp. 2712–2719.

Gunasekar, Suriya, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, et al. (2023). "Textbooks Are All You Need." In: *arXiv preprint arXiv:2306.11644*.

Gururangan, Suchin, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A Smith (2018). "Annotation Artifacts in Natural Language Inference Data." In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 107–112.

Hafner, Danijar, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson (2019). "Learning latent dynamics for planning from pixels." In: *International conference on machine learning*. PMLR, pp. 2555–2565.

Hambardzumyan, Karen, Hrant Khachatrian, and Jonathan May (2021). "Warp: Word-level adversarial reprogramming." In: *arXiv preprint arXiv:2101.00121*.

Hashimoto, Tatsunori B., Hugh Zhang, and Percy Liang (2019). "Unifying Human and Statistical Evaluation for Natural Language Generation." In: *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Helm, Ruud Van der (2006). "Towards a clarification of probability, possibility and plausibility: how semantics could help futures practice to improve." In: *Foresight*.

Hendrycks, Dan, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt (2020). "Measuring Massive Multitask Language Understanding." In: *International Conference on Learning Representations*.

Hendrycks, Dan and Kevin Gimpel (2016). "Gaussian error linear units (gelus)." In: *arXiv preprint arXiv:1606.08415*.

Hilton, Benjamin (May 2022). *No, DALL-E doesn't have a secret language.(or at least, we haven't found one yet)This viral DALL-E thread has some pretty astounding claims. But maybe the reason they're so astounding is that, for the most part, they're not true. Thread (1/15) https://t.co/8F2WDp7lTK*. en. https://twitter.com/benjamin_hilton/status/1531780892972175361?lang=en. Accessed: 2023-7-6.

Hinton, Geoffrey E (2002). "Training products of experts by minimizing contrastive divergence." In: *Neural Computation* 14.8, pp. 1771–1800.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). "Long short-term memory." In: *Neural computation* 9.8, pp. 1735–1780.

Holland, John H (2014). *Complexity: A very short introduction*. OUP Oxford.

Holtzman, Ari, Jan Buys, Maxwell Forbes, Antoine Bosselut, David Golub, and Yejin Choi (2018). "Learning to Write with Cooperative Discriminators." In: *Proceedings of the Association for Computational Linguistics*.

Holtzman, Ari, Jan Buys, Maxwell Forbes, and Yejin Choi (2020). "The Curious Case of Neural Text Degeneration." In: *International Conference on Learning Representations*.

Holtzman, Ari, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer (2021). "Surface Form Competition: Why the Highest Probability Answer Isn't Always Right." In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7038–7051.

Holtzman, Ari, Peter West, and Luke Zettlemoyer (2023). "Generative Models as a Complex Systems Science: How can we make sense of large language model behavior?" In: *preprint*.

Ilharco, Gabriel, Rowan Zellers, Ali Farhadi, and Hannaneh Hajishirzi (2021). "Probing contextual language models for common ground with visual representations." In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5367–5377.

Inan, Hakan, Khashayar Khosravi, and Richard Socher (2017). "Tying Word Vectors and Word Classifiers: A Loss Framework for Language Modeling." In: *Proceedings of theInternational Conference on Learning Representations*. URL: https://arxiv.org/abs/1611.01462.

Jacovi, Alon and Yoav Goldberg (2020). "Towards Faithfully Interpretable NLP Systems: How Should We Define and Evaluate Faithfulness?" In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4198–4205.

Jaeger, Tim Florian (2006). "Redundancy and syntactic reduction in spontaneous speech." PhD thesis. Stanford University Stanford, CA.

Jain, Neel, Khalid Saifullah, Yuxin Wen, John Kirchenbauer, Manli Shu, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein (2023). "Bring Your Own Data! Self-Supervised Evaluation for Large Language Models." In: *arXiv preprint arXiv:2306.13651*.

Jia, Robin and Percy Liang (2017). "Adversarial Examples for Evaluating Reading Comprehension Systems." In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2021–2031.

Jiang, Zhengbao, Frank F. Xu, Jun Araki, and Graham Neubig (2020a). "How Can We Know What Language Models Know?" In: *Transactions of the Association for*

Computational Linguistics 8, pp. 423–438. DOI: 10.1162/tacl_a_00324. URL: https://www.aclweb.org/anthology/2020.tacl-1.28.

Jiang, Zhengbao, Frank F Xu, Jun Araki, and Graham Neubig (2020b). "How can we know what language models know?" In: *Transactions of the Association for Computational Linguistics* 8, pp. 423–438.

Jozefowicz, Rafal, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu (2016). "Exploring the Limits of Language Modeling." In: *arXiv preprint arXiv:1602.02410*.

Kandpal, Nikhil, Eric Wallace, and Colin Raffel (2022). "Deduplicating training data mitigates privacy risks in language models." In: *International Conference on Machine Learning*. PMLR, pp. 10697–10707.

Karpathy, Andrej, Justin Johnson, and Li Fei-Fei (June 2015). "Visualizing and Understanding Recurrent Networks." In: arXiv: 1506.02078 [cs.LG].

Keskar, Nitish Shirish, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher (2019). "Ctrl: A conditional transformer language model for controllable generation." In: *arXiv preprint arXiv:1909.05858*.

Khandelwal, Urvashi, Kevin Clark, Dan Jurafsky, and Lukasz Kaiser (2019). "Sample Efficient Text Summarization Using a Single Pre-Trained Transformer." In: *arXiv preprint arXiv:1905.08836*.

Khandelwal, Urvashi, He He, Peng Qi, and Dan Jurafsky (2018). "Sharp Nearby, Fuzzy Far Away: How Neural Language Models Use Context." In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 284–294.

Khashabi, Daniel, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi (2020). "Unifiedqa: Crossing format boundaries with a single qa system." In: *arXiv preprint arXiv:2005.00700*.

Kiddon, Chloé, Luke Zettlemoyer, and Yejin Choi (2016). "Globally Coherent Text Generation with Neural Checklist Models." In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 329–339. URL: https://aclweb.org/anthology/D16-1032.

Kiho, Lee (2023). *ChatGPT_DAN: ChatGPT DAN, Jailbreaks prompt*. en.

Kilgarriff, Adam (2005). "Language is never, ever, ever, random." In: *Corpus Linguistics and Linguistic Theory* 12, p. 263275.

Kingma, Diederik P. and Jimmy Ba (2015). "Adam: A Method for Stochastic Optimization." In: *Proceedings of the International Conference on Learning Representations*. URL: http://arxiv.org/abs/1412.6980.

Klein, Guillaume, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush (2017). "OpenNMT: Open-Source Toolkit for Neural Machine Translation." In: *Proceedings of the Association of Computational Linguistics*. DOI: 10.18653/v1/P17-4012. URL: https://doi.org/10.18653/v1/P17-4012.

Koehn, Philipp and Rebecca Knowles (2017). "Six Challenges for Neural Machine Translation." In: *ACL 2017*, p. 28.

Koehn, Philipp, Franz Josef Och, and Daniel Marcu (2003). "Statistical Phrase-based Translation." In: *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume*

1. Edmonton, Canada: Association for Computational Linguistics, pp. 48–54. DOI: 10.3115/1073445.1073462. URL: https://doi.org/10.3115/1073445.1073462.

Krause, Jonathan, Justin Johnson, Ranjay Krishna, and Li Fei-Fei (2017). "A Hierarchical Approach for Generating Descriptive Image Paragraphs." In: *Proceedings of the Conference on Computer Vision and Pattern Recognition*.

Kulikov, Ilya, Alexander H Miller, Kyunghyun Cho, and Jason Weston (2018). "Importance of a Search Strategy in Neural Dialogue Modelling." In: *arXiv preprint arXiv:1811.00907*.

Lai, Guokun, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy (2017). "RACE: Large-scale ReAding Comprehension Dataset From Examinations." In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 785–794.

Lakretz, Yair, German Kruszewski, Theo Desbordes, Dieuwke Hupkes, Stanislas Dehaene, and Marco Baroni (June 2019). "The emergence of number and syntax units in LSTM language models." In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 11–20. DOI: 10.18653/v1/N19-1002. URL: https://aclanthology.org/N19-1002.

Lazaridou, Angeliki and Marco Baroni (2020). "Emergent multi-agent communication in the deep learning era." In: *arXiv preprint arXiv:2006.02419*.

Le, Matthew, Apoorv Vyas, Bowen Shi, Brian Karrer, Leda Sari, Rashel Moritz, Mary Williamson, Vimal Manohar, Yossi Adi, Jay Mahadeokar, and Wei-Ning Hsu (June 2023). "Voicebox: Text-Guided Multilingual Universal Speech Generation at Scale." In: arXiv: 2306.15687 [eess.AS].

Lee, Katherine, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini (2022). "Deduplicating Training Data Makes Language Models Better." In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8424–8445.

Lenat, Douglas B (1995). "CYC: A large-scale investment in knowledge infrastructure." In: *Communications of the ACM* 38.11, pp. 33–38.

Levy, Roger (2018). "Communicative Efficiency, Uniform Information Density, and the Rational Speech Act Theory." In: *CogSci*.

Levy, Roger and T Florian Jaeger (2007). "Speakers optimize information density through syntactic reduction." In: *Advances in neural information processing systems* 19, p. 849.

Li, Belinda Z, Jane Yu, Madian Khabsa, Luke Zettlemoyer, Alon Halevy, and Jacob Andreas (2021). "Quantifying Adaptability in Pre-trained Language Models with 500 Tasks." In: *arXiv preprint arXiv:2112.03204*.

Li, Jiwei, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan (2016a). "A Diversity-Promoting Objective Function for Neural Conversation Models." In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 110–119.

Li, Jiwei, Will Monroe, and Dan Jurafsky (2016). "A Simple, Fast Diverse Decoding Algorithm for Neural Generation." In: *CoRR* abs/1611.08562.

Li, Jiwei, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao (2016b). "Deep Reinforcement Learning for Dialogue Generation." In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1192–1202.

– (2016c). "Deep Reinforcement Learning for Dialogue Generation." In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, pp. 1192–1202. URL: https://aclweb.org/anthology/D16-1127.

Li, Xin and Dan Roth (2002). "Learning question classifiers." In: *COLING 2002: The 19th International Conference on Computational Linguistics*.

Liao, Thomas, Rohan Taori, Inioluwa Deborah Raji, and Ludwig Schmidt (2021). "Are we learning yet? a meta review of evaluation failures across machine learning." In: *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Lin, Xi Victoria, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, et al. (2021). "Few-shot learning with multilingual language models." In: *arXiv preprint arXiv:2112.10668*.

Linzen, Tal, Grzegorz Chrupała, Yonatan Belinkov, and Dieuwke Hupkes, eds. (Aug. 2019). *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Florence, Italy: Association for Computational Linguistics. URL: https://aclanthology.org/W19-4800.

Lipton, Zachary C (2018). "The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery." In: *Queue* 16.3, pp. 31–57.

Lipton, Zachary Chase, Sharad Vikram, and Julian McAuley (2015). "Capturing Meaning in Product Reviews with Character-Level Generative Text Models." In: *CoRR* abs/1511.03683. arXiv: 1511.03683. URL: http://arxiv.org/abs/1511.03683.

Liu, Chia-Wei, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau (2016). "How NOT To Evaluate Your Dialogue System: An Empirical Study of Unsupervised Evaluation Metrics for Dialogue Response Generation." In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, pp. 2122–2132. URL: https://aclweb.org/anthology/D16-1230.

Liu, Nelson F, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang (July 2023a). "Lost in the Middle: How Language Models Use Long Contexts." In: arXiv: 2307.03172 [cs.CL].

Liu, Pengfei, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig (2021). "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing." In: *arXiv preprint arXiv:2107.13586*.

Liu, Yang, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu (2023b). "Gpteval: Nlg evaluation using gpt-4 with better human alignment." In: *arXiv preprint arXiv:2303.16634*.

Lu, Yao, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp (2021). "Fantastically Ordered Prompts and Where to Find Them: Overcoming Few-Shot Prompt Order Sensitivity." In: *arXiv preprint arXiv:2104.08786*.

Luo, Zhengxiong, Dayou Chen, Yingya Zhang, Yan Huang, Liang Wang, Yujun Shen, Deli Zhao, Jingren Zhou, and Tieniu Tan (2023). "VideoFusion: Decomposed Diffusion Models for High-Quality Video Generation." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Luong, Thang, Hieu Pham, and Christopher D Manning (2015). "Effective Approaches to Attention-based Neural Machine Translation." In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1412–1421.

Manning, Christopher D, Kevin Clark, John Hewitt, Urvashi Khandelwal, and Omer Levy (2020). "Emergent linguistic structure in artificial neural networks trained by self-supervision." In: *Proceedings of the National Academy of Sciences* 117.48, pp. 30046–30054.

Mao, Huanru Henry, Bodhisattwa Prasad Majumder, Julian McAuley, and Garrison Cottrell (2019). "Improving Neural Story Generation by Targeted Common Sense Grounding." In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 5990–5995.

Marcus, Mitchell P, Mary Ann Marcinkiewicz, and Beatrice Santorini (1993). "Building a Large Annotated Corpus of English: The Penn Treebank." In: *Computational Linguistics* 19.2, pp. 313–330.

McCoy, R Thomas, Paul Smolensky, Tal Linzen, Jianfeng Gao, and Asli Celikyilmaz (2023). "How much do language models copy from their training data? evaluating linguistic novelty in text generation using raven." In: *Transactions of the Association for Computational Linguistics* 11, pp. 652–670.

McCoy, Tom, Ellie Pavlick, and Tal Linzen (July 2019). "Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference." In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 3428–3448. DOI: 10.18653/v1/P19-1334. URL: https://aclanthology.org/P19-1334.

Merity, Stephen, Nitish Shirish Keskar, and Richard Socher (2018). "Regularizing and Optimizing LSTM Language Models." In: *ICLR*.

Mihaylov, Todor, Peter Clark, Tushar Khot, and Ashish Sabharwal (2018). "Can a Suit of Armor Conduct Electricity? A New Dataset for Open Book Question Answering." In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2381–2391.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean (2013). "Distributed representations of words and phrases and their compositionality." In: *Advances in neural information processing systems* 26.

Mishra, Swaroop, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi (2022). "Cross-Task Generalization via Natural Language Crowdsourcing Instructions." In: *60th Annual Meeting of the Association for Computational Linguistics, ACL 2022*. Association for Computational Linguistics (ACL), pp. 3470–3487.

Mittal, Saurabh, Saikou Diallo, and Andreas Tolk (2018). *Emergent behavior in complex systems engineering: A modeling and simulation approach*. John Wiley & Sons.

Mori, Masahiro (June 2012). *The Uncanny Valley: The Original Essay by Masahiro Mori*. en. https://spectrum.ieee.org/the-uncanny-valley. Accessed: 2023-7-6.

Morin, Miguel and Matthew Willetts (2020). "Non-determinism in tensorflow resnets." In: *arXiv preprint arXiv:2001.11396*.

Mostafazadeh, Nasrin, Michael Roth, Annie Louis, Nathanael Chambers, and James Allen (2017). "Lsdsem 2017 shared task: The story cloze test." In: *Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics*, pp. 46–51.

Mou, Lili, Yiping Song, Rui Yan, Ge Li, Lu Zhang, and Zhi Jin (2016). "Sequence to Backward and Forward Sequences: A Content-Introducing Approach to Generative Short-Text Conversation." In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 3349–3358.

Mukherjee, Subhabrata, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah (June 2023). "Orca: Progressive Learning from Complex Explanation Traces of GPT-4." In: arXiv: 2306.02707 [cs.CL].

Müller, Mathias, Annette Rios Gonzales, and Rico Sennrich (2020). "Domain Robustness in Neural Machine Translation." In: *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pp. 151–164.

Murphy, Kevin P. (2012). *Machine Learning: A Probabilistic Perspective*. The MIT Press. ISBN: 0262018020.

Nallapati, Ramesh, Bowen Zhou, Cicero dos Santos, Caglar Gulcehre, and Bing Xiang (2016). "Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond." In: *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pp. 280–290. URL: https://www.aclweb.org/anthology/K16-1028.

Newman, MEJ (2011). "Resource letter cs–1: Complex systems." In: *American Journal of Physics* 79.8, pp. 800–810.

Nie, Yixin, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela (2020). "Adversarial NLI: A New Benchmark for Natural Language Understanding." In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4885–4901.

Novikova, Jekaterina, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser (2017). "Why We Need New Evaluation Metrics for NLG." In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 2241–2252. URL: https://www.aclweb.org/anthology/D17-1238.

Och, Franz Josef (2003). "Minimum Error Rate Training in Statistical Machine Translation." In: *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*. Sapporo, Japan: Association for Computational Linguistics, pp. 160–167. DOI: 10.3115/1075096.1075117. URL: http://www.aclweb.org/anthology/P03-1021.

Olah, Chris (Aug. 2015). *Understanding LSTM Networks*. en. https://colah.github.io/posts/2015-08-Understanding-LSTMs/. Accessed: 2023-7-7.

– (June 2022). *Mechanistic Interpretability, Variables, and the Importance of Interpretable Bases*. en. https://transformer-circuits.pub/2022/mech-interp-essay/index.html. Accessed: 2023-7-8.

Olsson, Catherine, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah (2022). "In-context Learning and Induction Heads." In: *Transformer Circuits Thread*. URL: https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html.

Oluwatobi, Olabiyi and Erik Mueller (July 2020). "DLGNet: A Transformer-based Model for Dialogue Response Generation." In: *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*. Online: Association for Computational Linguistics, pp. 54–62. DOI: 10.18653/v1/2020.nlp4convai-1.7. URL: https://www.aclweb.org/anthology/2020.nlp4convai-1.7.

OpenAI (July 2023). *GPT-4 API general availability and deprecation of older models in the Completions API*. en. https://openai.com/blog/gpt-4-api-general-availability. Accessed: 2023-7-6.

OpenAI (Mar. 2023). "GPT-4 Technical Report." In: arXiv: 2303.08774 [cs.CL].

Ouyang, Long, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. (2022). "Training language models to follow instructions with human feedback." In: *Advances in Neural Information Processing Systems* 35, pp. 27730–27744.

Pantel, Patrick, Rahul Bhagat, Bonaventura Coppola, Timothy Chklovski, and Eduard Hovy (Apr. 2007). "ISP: Learning Inferential Selectional Preferences." In: *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*. Rochester, New York: Association for Computational Linguistics, pp. 564–571. URL: https://www.aclweb.org/anthology/N07-1071.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu (2002). "BLEU: a method for automatic evaluation of machine translation." In: *Proceedings of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 311–318. URL: https://www.aclweb.org/anthology/P02-1040.

Parcalabescu, Letitia, Albert Gatt, Anette Frank, and Iacer Calixto (2021). "Seeing past words: Testing the cross-modal capabilities of pretrained V&L models on counting tasks." In: *Proceedings of the 1st Workshop on Multimodal Semantic Representations (MMSR)*, pp. 32–44.

Parikh, Ankur, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit (2016). "A Decomposable Attention Model for Natural Language Inference." In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Asso-

ciation for Computational Linguistics, pp. 2249–2255. DOI: 10.18653/v1/D16-1244. URL: http://www.aclweb.org/anthology/D16-1244.

Pascanu, Razvan, Tomas Mikolov, and Yoshua Bengio (2013). "On the Difficulty of Training Recurrent Neural Networks." In: *International Conference on Machine Learning (ICML)*, pp. 1310–1318.

Patel, Roma and Ellie Pavlick (Nov. 2021). ""Was it "stated" or was it "claimed"?: How linguistic bias affects generative language models." In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 10080–10095. DOI: 10.18653/v1/2021.emnlp-main.790. URL: https://aclanthology.org/2021.emnlp-main.790.

Paulus, Romain, Caiming Xiong, and Richard Socher (2018). "A Deep Reinforced Model for Abstractive Summarization." In: *CoRR* abs/1705.04304.

Peng, Nanyun, Marjan Ghazvininejad, Jonathan May, and Kevin Knight (June 2018). "Towards Controllable Story Generation." In: *Proceedings of the First Workshop on Storytelling*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 43–49. DOI: 10.18653/v1/W18-1505. URL: https://www.aclweb.org/anthology/W18-1505.

Pennington, Jeffrey, Richard Socher, and Christopher Manning (2014). "Glove: Global Vectors for Word Representation." In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Doha, Qatar: Association for Computational Linguistics, pp. 1532–1543. URL: http://www.aclweb.org/anthology/D14-1162.

Perez, Ethan, Douwe Kiela, and Kyunghyun Cho (2021). "True Few-Shot Learning with Language Models." In: *arXiv preprint arXiv:2105.11447*.

Perry, Alex (June 2023). *OpenAI updates GPT-4 with new features*. en. https://mashable.com/article/openai-chatgpt-gpt-4-function-calling-update. Accessed: 2023-7-6.

Peters, Ben, Vlad Niculae, and André FT Martins (2019). "Sparse Sequence-to-Sequence Models." In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1504–1519.

Peters, Matthew E., Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer (June 2018). "Deep Contextualized Word Representations." In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 2227–2237. DOI: 10.18653/v1/N18-1202. URL: https://aclanthology.org/N18-1202.

Petroni, Fabio, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller (2019). "Language Models as Knowledge Bases?" In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2463–2473.

Piantadosi, Steven T (2014). "Zipf's word frequency law in natural language: A critical review and future directions." In: *Psychonomic bulletin & review* 21.5, pp. 1112–1130.

Pichai, Sundar (Feb. 2023). *An important next step on our AI journey*. en. https://blog.google/technology/ai/bard-google-ai-search-updates/. Accessed: 2023-7-6.

Poliak, Adam, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme (2018). "Hypothesis Only Baselines in Natural Language Inference." In: *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pp. 180–191.

Press, Ofir, Noah A Smith, and Mike Lewis (2021). "Shortformer: Better Language Modeling using Shorter Inputs." In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 5493–5505.

Prinz, Wolfgang (2006). "Messung kontra Augenschein." In: *Psychologische Rundschau* 57.2, pp. 106–111.

Provilkov, Ivan, Dmitrii Emelianenko, and Elena Voita (2020). "BPE-Dropout: Simple and Effective Subword Regularization." In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1882–1892.

Radford, Alec, Rafal Jozefowicz, and Ilya Sutskever (2017). "Learning to generate reviews and discovering sentiment." In: *arXiv preprint arXiv:1704.01444*.

Radford, Alec, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever (2018a). "Improving language understanding by generative pre-training." In.

– (2018b). *Improving language understanding by generative pre-training*. Unpublished manuscript. URL: https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf.

Radford, Alec, Jeffrey Wu, Dario Amodei, Daniela Amodei, Jack Clark, Miles Brundage, and Ilya Sutskever (2019a). "Better language models and their implications." In: *OpenAI Blog* 1, p. 2.

Radford, Alec, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever (2019b). "Language models are unsupervised multitask learners." In: *OpenAI Blog*.

Radford, Alec, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. (2019c). "Language models are unsupervised multitask learners." In: *OpenAI blog* 1.8, p. 9.

Ray, Partha Pratim (2023). "ChatGPT: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope." In: *Internet of Things and Cyber-Physical Systems*.

Reynolds, Laria and Kyle McDonell (2021). "Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm." In: *arXiv preprint arXiv:2102.07350*.

Ribeiro, Marco Tulio, Tongshuang Wu, Carlos Guestrin, and Sameer Singh (2020). "Beyond Accuracy: Behavioral Testing of NLP Models with CheckList." In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4902–4912.

Roemmele, Melissa, Cosmin Adrian Bejan, and Andrew S Gordon (2011). "Choice of Plausible Alternatives: An Evaluation of Commonsense Causal Reasoning." In: *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*.

Rogers, Anna, Olga Kovaleva, and Anna Rumshisky (2020). "A primer in bertology: What we know about how bert works." In: *Transactions of the Association for Computational Linguistics* 8, pp. 842–866.

Rohrbach, Anna, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko (2018). "Object Hallucination in Image Captioning." In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 4035–4045.

Rombach, Robin, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer (2021). *High-Resolution Image Synthesis with Latent Diffusion Models*. arXiv: 2112.10752 [cs.CV].

Romera-Paredes, Bernardino and Philip Torr (2015). "An embarrassingly simple approach to zero-shot learning." In: *International conference on machine learning*. PMLR, pp. 2152–2161.

Rosenthal, Robert (1976). "Experimenter effects in behavioral research." In.

Rudman, William and Carsten Eickhoff (2023). "Stable Anisotropic Regularization." In: *arXiv preprint arXiv:2305.19358*.

Schick, Timo and Hinrich Schütze (2020a). "Exploiting cloze questions for few-shot text classification and natural language inference." In: *arXiv preprint arXiv:2001.07676*.

– (2020b). "Few-shot text generation with pattern-exploiting training." In: *arXiv preprint arXiv:2012.11926*.

– (2020c). "It's Not Just Size That Matters: Small Language Models Are Also Few-Shot Learners." In: *arXiv preprint arXiv:2009.07118*.

Schmaltz, Allen, Alexander M. Rush, and Stuart Shieber (2016). "Word Ordering Without Syntax." In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, pp. 2319–2324. URL: https://aclweb.org/anthology/D16-1255.

Schulman, John, Barret Zoph, Christina Kim, Jacob Hilton, Jacob Menick, Jiayi Weng, Juan Felipe Ceron Uribe, Liam Fedus, Luke Metz, Michael Pokorny, Rapha Gontijo Lopes, Shengjia Zhao, Arun Vijayvergiya, Eric Sigler, Adam Perelman, Chelsea Voss, Mike Heaton, Joel Parish, Dave Cummings, Rajeev Nayak, Valerie Balcom, David Schnurr, Tomer Kaftan, Chris Hallacy, Nicholas Turley, Noah Deutsch, Vik Goel, Jonathan Ward, Aris Konstantinidis, Wojciech Zaremba, Long Ouyang, Leonard Bogdonoff, Joshua Gross, David Medina, Sarah Yoo, Teddy Lee, Ryan Lowe, Dan Mossing, Joost Huizinga, Roger Jiang, Carroll Wainwright, Diogo Almeida, Steph Lin, Marvin Zhang, Kai Xiao, Katarina Slama, Steven Bills, Alex Gray, Jan Leike, Jakub Pachocki, Phil Tillet, Shantanu Jain, Greg Brockman, Nick Ryder, Alex Paino, Qiming Yuan, Clemens Winter, Ben Wang, Mo Bavarian, Igor Babuschkin, Szymon Sidor, Ingmar Kanitscheider, Mikhail Pavlov, Matthias Plappert, Nik Tezak, Heewoo Jun, William Zhuk, Vitchyr Pong, Lukasz Kaiser, Jerry Tworek, Andrew Carr, Lilian Weng, Sandhini Agarwal, Karl Cobbe, Vineet Kosaraju, Alethea Power, Stanislas Polu, Jesse Han, Raul Puri, Shawn Jain, Benjamin Chess, Christian Gibson, Oleg Boiko, Emy Parparita, Amin Tootoonchian, Kyle Kosic, and Christopher Hesse (2022). "Introducing ChatGPT." In: *OpenAI Blog*. URL: https://openai.com/blog/chatgpt.

See, Abigail, Peter J Liu, and Christopher D Manning (2017). "Get To The Point: Summarization with Pointer-Generator Networks." In: *Proceedings of the 55th Annual*

Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1073–1083.

See, Abigail, Aneesh Pappu, Rohun Saxena, Akhila Yerukola, and Christopher D Manning (2019). "Do Massively Pretrained Language Models Make Better Storytellers?" In: *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pp. 843–861.

Semeniuta, Stanislau, Aliaksei Severyn, and Sylvain Gelly (2018). "On accurate evaluation of gans for language generation." In: *arXiv preprint arXiv:1806.04936*.

Sennrich, Rico, Barry Haddow, and Alexandra Birch (2016). "Neural Machine Translation of Rare Words with Subword Units." In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1715–1725.

Shao, Yuanlong, Stephan Gouws, Denny Britz, Anna Goldie, Brian Strope, and Ray Kurzweil (2017). "Generating High-Quality and Informative Conversation Responses with Sequence-to-Sequence Models." In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 2210–2219. URL: http://aclweb.org/anthology/D17-1235.

Shen, Tianxiao, Tao Lei, Regina Barzilay, and Tommi Jaakkola (2017). "Style transfer from non-parallel text by cross-alignment." In: *Advances in neural information processing systems*, pp. 6830–6841.

Shi, Xing, Yijun Xiao, and Kevin Knight (2020). "Why neural machine translation prefers empty outputs." In: *arXiv preprint arXiv:2012.13454*.

Shin, Taylor, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh (Nov. 2020). "AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts." In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, pp. 4222–4235. DOI: 10.18653/v1/2020.emnlp-main.346. URL: https://www.aclweb.org/anthology/2020.emnlp-main.346.

Socher, Richard, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts (2013). "Recursive deep models for semantic compositionality over a sentiment treebank." In: *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642.

Southern, Matt G (Jan. 2023). *[No title]*. https://www.searchenginejournal.com/openai-chatgpt-update/476116/. Accessed: 2023-7-6.

Speer, Robyn, Joshua Chin, and Catherine Havasi (2017). "Conceptnet 5.5: An open multilingual graph of general knowledge." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 31.

Srivastava, Nitish, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov (2014). "Dropout: A Simple Way to Prevent Neural Networks from Overfitting." In: *Journal of Machine Learning Research* 15.1, pp. 1929–1958.

Stability AI, company (2023). *StableLM: StableLM: Stability AI Language Models*. en.

Stahlberg, Felix and Bill Byrne (Nov. 2019a). "On NMT Search Errors and Model Errors: Cat Got Your Tongue?" In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference*

on Natural Language Processing (EMNLP-IJCNLP). Hong Kong, China: Association for Computational Linguistics, pp. 3356–3362. DOI: 10.18653/v1/D19-1331. URL: https://aclanthology.org/D19-1331.

Stahlberg, Felix and Bill Byrne (2019b). "On NMT Search Errors and Model Errors: Cat Got Your Tongue?" In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3356–3362.

Stahlberg, Felix, Ilia Kulikov, and Shankar Kumar (2022). "Uncertainty Determines the Adequacy of the Mode and the Tractability of Decoding in Sequence-to-Sequence Models." In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8634–8645.

Steinert-Threlkeld, Shane, Xuhui Zhou, Zeyu Liu, and C M Downey (Mar. 2022). "Emergent Communication Fine-tuning (EC-FT) for Pretrained Language Models." In: *ICLR 2022 EmeCom Workshop*.

Sun, Simeng, Kalpesh Krishna, Andrew Mattarella-Micke, and Mohit Iyyer (2021). "Do Long-Range Language Models Actually Use Long-Range Context?" In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 807–822.

Sundermeyer, Martin, Ralf Schlüter, and Hermann Ney (2012). "LSTM neural networks for language modeling." In: *Thirteenth annual conference of the international speech communication association*.

Sutskever, Ilya, Oriol Vinyals, and Quoc VV Le (2014). "Sequence to Sequence Learning with Neural Networks." In: *Advances in Neural Information Processing Systems*, pp. 3104–3112.

Talmor, Alon, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant (2019). "CommonsenseQA: A Question Answering Challenge Targeting Commonsense Knowledge." In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4149–4158.

Tang, Jianheng, Tiancheng Zhao, Chenyan Xiong, Xiaodan Liang, Eric Xing, and Zhiting Hu (2019). "Target-Guided Open-Domain Conversation." In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5624–5634.

Taori, Rohan, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto (2023). *Stanford Alpaca: An Instruction-following LLaMA model*. https://github.com/tatsu-lab/stanford_alpaca.

Teehan, Ryan, Miruna Clinciu, Oleg Serikov, Eliza Szczechla, Natasha Seelam, Shachar Mirkin, and Aaron Gokaslan (2022). "Emergent Structures and Training Dynamics in Large Language Models." In: *Proceedings of BigScience Episode\# 5–Workshop on Challenges & Perspectives in Creating Large Language Models*, pp. 146–159.

Tenney, Ian, Dipanjan Das, and Ellie Pavlick (2019). "BERT Rediscovers the Classical NLP Pipeline." In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4593–4601.

Tevet, Guy, Gavriel Habib, Vered Shwartz, and Jonathan Berant (2018). "Evaluating Text GANs as Language Models." In: *CoRR* abs/1810.12686. arXiv: 1810.12686. URL: http://arxiv.org/abs/1810.12686.

Tirumala, Kushal, Aram Markosyan, Luke Zettlemoyer, and Armen Aghajanyan (2022). "Memorization without overfitting: Analyzing the training dynamics of large language models." In: *Advances in Neural Information Processing Systems* 35, pp. 38274–38290.

Tomasello, Michael (2005). *Constructing a language: A usage-based theory of language acquisition*. Harvard university press.

Touvron, Hugo, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample (Feb. 2023). "LLaMA: Open and Efficient Foundation Language Models." In: arXiv: 2302.13971 [cs.CL].

Turpin, Miles, Julian Michael, Ethan Perez, and Samuel R Bowman (2023). "Language Models Don't Always Say What They Think: Unfaithful Explanations in Chain-of-Thought Prompting." In: *arXiv preprint arXiv:2305.04388*.

Ullman, Tomer (2023). "Large language models fail on trivial alterations to theory-of-mind tasks." In: *arXiv preprint arXiv:2302.08399*.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin (2017). "Attention is all you need." In: *Advances in Neural Information Processing Systems*, pp. 5998–6008.

Vedantam, Ramakrishna, C Lawrence Zitnick, and Devi Parikh (2015). "Cider: Consensus-based image description evaluation." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4566–4575.

Venigalla, Abhinav and Linden Li (Sept. 2022). *Mosaic LLMs (Part 2): GPT-3 quality for <$500k*. https://www.mosaicml.com/blog/gpt-3-quality-for-500k. Accessed: 2023-7-7.

Vicuna Team, The (n.d.). *Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality*. en. https://lmsys.org/blog/2023-03-30-vicuna/. Accessed: 2023-7-6.

Vijayakumar, Ashwin K., Michael Cogswell, Ramprasaath R. Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra (2018). "Diverse beam search for improved description of complex scenes." In: *Thirty-Second AAAI Conference on Artificial Intelligence*.

Vijayakumar, Ashwin K, Michael Cogswell, Ramprasath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra (2016). "Diverse Beam Search: Decoding Diverse Solutions from Neural Sequence Models." In: *arXiv preprint arXiv:1610.02424*.

Wallace, Eric, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh (2019). "Universal Adversarial Triggers for Attacking and Analyzing NLP." In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2153–2162.

Wang, Alex, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman (2019). "SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems." In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Vol. 32. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper/2019/file/4496bf24afe7fab6f046bf4923da8de6-Paper.pdf.

Wang, Alex, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman (2018). "GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding." In: *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 353–355.

Wang, Changhan, Kyunghyun Cho, and Jiatao Gu (2020). "Neural machine translation with byte-level subwords." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34, pp. 9154–9160.

Wang, Chaojun and Rico Sennrich (2020). "On Exposure Bias, Hallucination and Domain Shift in Neural Machine Translation." In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 3544–3552.

Wang, Hongning, Yue Lu, and ChengXiang Zhai (2010). "Latent Aspect Rating Analysis on Review Text Data: A Rating Regression Approach." In: *SIGKDD Conference on Knowledge Discovery and Data Mining*.

Wang, Lingxiao, Jing Huang, Kevin Huang, Ziniu Hu, Guangtao Wang, and Quanquan Gu (2020). "Improving neural language generation with spectrum control." In: *International Conference on Learning Representations*.

Wang, Su, Greg Durrett, and Katrin Erk (June 2018). "Modeling Semantic Plausibility by Injecting World Knowledge." In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 303–308. DOI: 10.18653/v1/N18-2049. URL: https://www.aclweb.org/anthology/N18-2049.

Warstadt, Alex, Leshem Choshen, Aaron Mueller, Adina Williams, Ethan Wilcox, and Chengxu Zhuang (2023). "Call for Papers–The BabyLM Challenge: Sample-efficient pretraining on a developmentally plausible corpus." In: *arXiv preprint arXiv:2301.11796*.

Watanabe, Taro, Jun Suzuki, Hajime Tsukada, and Hideki Isozaki (2007). "Online Large-Margin Training for Statistical Machine Translation." In: *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. Prague, Czech Republic: Association for Computational Linguistics, pp. 764–773. URL: http://www.aclweb.org/anthology/D/D07/D07-1080.

Wei, Jason, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le (2021). "Finetuned Language Models Are Zero-Shot Learners." In: *arXiv preprint arXiv:2109.01652*.

Wei, Jason, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori

Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus (2022). "Emergent Abilities of Large Language Models." In: *arXiv preprint arXiv:2206.07682*.

Weiss, Gail, Yoav Goldberg, and Eran Yahav (2021). "Thinking like transformers." In: *International Conference on Machine Learning*. PMLR, pp. 11080–11090.

Welleck, Sean, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston (2019). "Neural Text Generation With Unlikelihood Training." In: *International Conference on Learning Representations*.

Williams, Adina, Nikita Nangia, and Samuel R. Bowman (2017). "A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference." In: *CoRR* abs/1704.05426. arXiv: 1704.05426. URL: http://arxiv.org/abs/1704.05426.

Wilson, Andrew (2023). *How to Jailbreak ChatGPT to Unlock its Full Potential*. https://approachableai.com/how-to-jailbreak-chatgpt/. Accessed: 2023-7-6.

Wiseman, Sam, Stuart Shieber, and Alexander Rush (2017). "Challenges in Data-to-Document Generation." In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 2253–2263. URL: https://www.aclweb.org/anthology/D17-1239.

Wolf, Thomas, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush (Oct. 2020). "Transformers: State-of-the-Art Natural Language Processing." In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, pp. 38–45. DOI: 10.18653/v1/2020.emnlp-demos.6. URL: https://www.aclweb.org/anthology/2020.emnlp-demos.6.

Wolpert, D H and W G Macready (Apr. 1997). "No free lunch theorems for optimization." In: *IEEE Trans. Evol. Comput.* 1.1, pp. 67–82.

Wu, Yonghui, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean (2016). "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation." In: *CoRR* abs/1609.08144. arXiv: 1609.08144. URL: http://arxiv.org/abs/1609.08144.

Xie, Zhuohan, Trevor Cohn, and Jey Han Lau (Jan. 2023). "Can Very Large Pretrained Language Models Learn Storytelling With A Few Examples?" In: arXiv: 2301.09790 [cs.CL].

Xu, Jingjing, Xuancheng Ren, Junyang Lin, and Xu Sun (2018). "Diversity-Promoting GAN: A Cross-Entropy Based Generative Adversarial Network for Diversified Text Generation." In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium, pp. 3940–3949. URL: http://www.aclweb.org/anthology/D18-1428.

Yang, Jingfeng, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Bing Yin, and Xia Hu (2023). "Harnessing the power of llms in practice: A survey on chatgpt and beyond." In: *arXiv preprint arXiv:2304.13712*.

Yao, Lili, Yaoyuan Zhang, Yansong Feng, Dongyan Zhao, and Rui Yan (2017). "Towards implicit content-introducing for generative short-text conversation systems." In: *Proceedings of the 2017 conference on empirical methods in natural language processing*, pp. 2190–2199.

Ye, Qinyuan, Bill Yuchen Lin, and Xiang Ren (2021a). "CrossFit: A Few-shot Learning Challenge for Cross-task Generalization in NLP." In: *arXiv preprint arXiv:2104.08835*.

– (Nov. 2021b). "CrossFit: A Few-shot Learning Challenge for Cross-task Generalization in NLP." In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 7163–7189. DOI: 10.18653/v1/2021.emnlp-main.572. URL: https://aclanthology.org/2021.emnlp-main.572.

Yu, Lantao, Weinan Zhang, Jun Wang, and Yong Yu (2017a). "SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient." In: *Proceedings of the Association for the Advancement of Artificial Intelligence*, pp. 2852–2858.

Yu, Lei, Phil Blunsom, Chris Dyer, Edward Grefenstette, and Tomas Kocisky (2017b). "The Neural Noisy Channel." In: *International Conference on Learning Representations*.

Yu, Licheng, Hao Tan, Mohit Bansal, and Tamara L Berg (2017c). "A Joint Speaker-Listener-Reinforcer Model for Referring Expressions." In: *Proceedings of the Conference on Computer Vision and Pattern Recognition*. Vol. 2.

Yu, Lili, Dániel Simig, Colin Flaherty, Armen Aghajanyan, Luke Zettlemoyer, and Mike Lewis (2023). "Megabyte: Predicting million-byte sequences with multiscale transformers." In: *arXiv preprint arXiv:2305.07185*.

Zellers, Rowan, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi (2019a). "HellaSwag: Can a Machine Really Finish Your Sentence?" In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800.

Zellers, Rowan, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi (2019b). "Defending against neural fake news." In: *Advances in neural information processing systems* 32.

Zhang, Chaoning, Chenshuang Zhang, Chenghao Li, Yu Qiao, Sheng Zheng, Sumit Kumar Dam, Mengchun Zhang, Jung Uk Kim, Seong Tae Kim, Jinwoo Choi, et al. (2023). "One small step for generative ai, one giant leap for agi: A complete survey on chatgpt in aigc era." In: *arXiv preprint arXiv:2304.06488*.

Zhang, Xiang, Junbo Zhao, and Yann LeCun (2015). "Character-level Convolutional Networks for Text Classification." In: *Advances in Neural Information Processing Systems* 28, pp. 649–657.

Zhao, Tony Z, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh (2021). "Calibrate Before Use: Improving Few-Shot Performance of Language Models." In: *arXiv preprint arXiv:2102.09690*.

Zheng, Lianmin, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P Xing, Hao Zhang, Joseph E Gonzalez,

and Ion Stoica (June 2023). "Judging LLM-as-a-judge with MT-Bench and Chatbot Arena." In: arXiv: 2306.05685 [cs.CL].

Zhong, Ruiqi, Kristy Lee, Zheng Zhang, and Dan Klein (2021). "Adapting Language Models for Zero-shot Learning by Meta-tuning on Dataset and Prompt Collections." In: *arXiv preprint arXiv:2109.01652*.

Zhou, Chunting, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. (2023). "Lima: Less is more for alignment." In: *arXiv preprint arXiv:2305.11206*.

Zhou, Kun, Kai Zhang, Yu Wu, Shujie Liu, and Jingsong Yu (2019). "Unsupervised Context Rewriting for Open Domain Conversation." In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 1834–1844.

Zhu, Yaoming, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu (2018). "Texygen: A Benchmarking Platform for Text Generation Models." In: *ACM SIGIR Conference on Research and Development in Information Retrieval*.

Zhu, Yukun, Ryan Kiros, Richard Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler (2015). "Aligning Books and Movies: Towards Story-like Visual Explanations by Watching Movies and Reading Books." In: *arXiv preprint arXiv:1506.06724*.

Ziegler, Daniel M, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving (2019). "Fine-tuning language models from human preferences." In: *arXiv preprint arXiv:1909.08593*.

百度百科 (2022). 普通高等学校招生全国统一考试. URL: https://baike.baidu.com/item/\%E6\%99\%AE\%E9\%80\%9A\%E9\%AB\%98\%E7\%AD\%89\%E5\%AD\%A6\%E6\%A0\%A1\%E6\%8B\%9B\%E7\%94\%9F\%E5\%85\%A8\%E5\%9B\%BD\%E7\%BB\%9F\%E4\%B8\%80\%E8\%80\%83\%E8\%AF\%95/2567351?fromtitle=\%E9\%AB\%98\%E8\%80\%83\&fromid=219910.

COLOPHON

This dissertation was typeset using the typographical look-and-feel `classicthesis`, developed by André Miede and Ivo Pletikosić, and further refined by Amrita Mazumdar, with some imprvoements added by Maxwell Forbes.

Final Version as of August 9, 2023 (classicthesis v4.6).