Calculemus

The famous philosopher and mathematician Leibniz cherished the hope that one day, if a dispute arose between two philosophers, "it would suffice for them to take their pencils in their hands and to sit down at the abacus, and say to each other... Let us calculate (*Calculemus*)" [1]. For this reason, pioneers of AI like Terry Winograd consider their work as heirs to a long tradition from philosophers like Leibniz, Descartes, and Hobbes [2]. These philosophers laid out a key motivation for AI: answering questions that people cannot.

How close is AI to achieving the philosopher's ambition? Supervised machine learning excels at imitating human decision-making. For example, neural networks achieve parity with humans at answering factoid questions about Wikipedia passages [3]. Unfortunately, supervised learning cannot train models to solve tasks that people cannot, because supervised learning relies on human-provided labels. We need to develop learning paradigms that generalize beyond the available supervision. To answer questions that people cannot, machine learning should generalize from questions that people can answer to questions that are much harder for people to answer. My research aims to improve generalization in machine learning, with a special interest in question-answering (QA) as a real-world, natural language application. Below, I outline three research directions that work towards improving generalization in QA, and I describe my work in each direction.

1 Debate

Debate is the process of defending an answer to a question with arguments and evidence. Some have proposed to train QA agents to debate, in order to surface the strongest arguments and counter-arguments for various answers [4, 5]. Such proposals hypothesize that strong arguments and counter-arguments make it easier for a QA model to evaluate an answer to a question. An answer alone can be challenging for people evaluate (i.e., "Yes, the U.S. should raise taxes"). It is much easier for people to evaluate an answer in light of key arguments that support or discredit the answer (i.e., "Yes, the U.S. should raise taxes because of these reasons..."). Importantly, it should be easier to justify correct answers than incorrect ones, in order for debate to help QA models correctly evaluate answers.

In my recent work [6], I tested debate empirically: does debate actually help in answering harder questions? We explored this question in the context of natural language QA. We found that a simple form of debate does indeed improve generalization in QA. In this form of debate, agents learn to quote evidence and counter-evidence from a source text. With agent-chosen evidence and counter-evidence, QA models can generalize to longer passages and harder questions than seen during training. People can even answer accurately questions about long passages using just the agent-chosen evidence (only ~20% of the source text). Even with our current, limited machine learning methods, training agents to debate helps with QA.

The above results are promising, but in our setup, an agent could only justify an answer using existing statements or evidence. Ultimately, we want agents to provide any necessary justification for their answers, including via novel, free-form text. To this end, I also worked on developing a task for Long-form QA called "Explain Like I'm Five," where questions require a free-form, paragraph-long justification ("Do lower interest rates increase investment? How?") [7]. In the context of debate, more expressive justifications can make it easier to evaluate the correctness of an answer. Down the line, I am interested in training models to debate in free-form natural language directly from human debates (e.g., from Reddit's "Change My View" forum [8] or `Kialo.com`). More expressive debating agents may provide better evidence or arguments for an answer. Better evidence makes it more likely that a QA model generalizes to harder questions in light of the presented evidence.

2 Decomposing Questions into Sub-Questions

Another approach to generalizing to harder questions is to break harder questions down into easier sub-questions. For example, current QA models can answer "single-hop" questions that only require reasoning over one piece of information (i.e., a single paragraph) [3, 9]. However, QA models struggle to answer "multi-hop" questions that require reasoning over multiple pieces of information (i.e., several paragraphs from different documents) [10]. To fix the issue, we could collect a large dataset of multi-hop questions and train QA models on the dataset. However, it is time-consuming for people to label questions with answers, especially multi-hop questions over several documents. Instead, we can learn to decompose multi-hop questions ("Who was born earlier, George Washington or Abraham Lincoln?") into single-hop questions ("When was George Washington born?" and "When was Abraham Lincoln born?"). By leveraging a model's ability to answer single-hop questions, the model can generalize to multi-hop questions simply by learning

to compose the answers to sub-questions. It is possible to recursively apply the process of decomposing questions to answer even harder questions; this procedure (termed "Iterated Amplification") holds promise in answering successively harder questions [11]. I am interested in understanding how to learn to decompose questions effectively in practice.

For example, one practical bottleneck is that one of the easiest ways to learn to decompose questions is via supervision – extra labor for people. Some forms of decomposition can require many example decompositions to learn (i.e., generating free-form sub-questions). It might also be challenging to collect enough decomposition supervision for a wide variety of questions. Moreover, the best decomposition can depend on what sub-questions a particular model can answer, but it is impractical to collect new decompositions for each model or as a model improves throughout training.

Thus, I am interested in exploring how effectively models can learn to decompose questions without any supervision. In particular, I plan to adapt unsupervised machine translation to decompose questions by leveraging massive databases of (unanswered) multi-hop and single-hop questions. Unsupervised machine translation has proven exceptionally effective at learning from unlabelled data. As a result, unsupervised translation often outperforms supervised translation for "low-resource" language pairs with few supervised translations (i.e., English to Nepali) [12, 13]. Furthermore, unsupervised decomposition can be easily adapted to semi-supervised decomposition when examples of decomposing questions are available. By adding sub-questions and their answers to a QA model's input, a QA model can learn to generalize better to multi-hop questions using fewer labeled answers.

3 Evaluating Generalization to Improve Generalization

In order to improve generalization, we need to appropriately evaluate generalization. There are several approaches for constructing out-of-distribution test sets to measure generalization. One approach is to have people create examples on which current models fail [14, 15]. When relying on people is expensive, we can also find examples in existing datasets where current models fail [16, 17, 18]. By constructing out-of-distribution test sets, I plan to directly train models to learn on supervised data in a way that generalizes to out-of-distribution data. In particular, I plan to use meta-learning to learn importance weights for every labeled example. We can increase or decrease an example's weight based on how generalization performance changes after a gradient descent step on the example. There is evidence that learning importance weights can improve in-distribution generalization; in program induction, [19] uses meta-learning to learn from a noisy program label only when learning would improve generalization on a validation set. In short, we can train models to be aware that they will be evaluated out-of-distribution, encouraging models to learn patterns that generalize better.

Conclusion

Admittedly, the above research directions are only the first steps towards AI that fulfills Leibniz's hope: answering questions that people cannot answer. To answer such questions, we will need to go far beyond methods which answer questions about interest rates in "Explain Like I'm 5." Leibniz asked questions such as "In what is morality rooted?" and "How can God exist if there is evil in the world?" These are questions whose answers people orient their lives around, questions whose answers which will require us to place an enormous trust in AI. By advancing generalization in QA, I aspire to one day ask a question I cannot answer and say "*Calculemus*" and, from an AI system, receive a response that I can stake my life on.

References

[1] Gottfried Leibniz. *1688*. De Arte Characteristica ad Perficiendas Scientias Ratione Nitentes.

[2] Terry Winograd. *Cambridge University Press, 1990*. Thinking Machines: Can There Be? Are We?

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. *NAACL 2019*. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

[4] Geoffrey Irving, Paul Christiano, Dario Amodei. *arXiv 2018*. AI Safety via Debate.

[5] Geoffrey Irving and Amanda Askell. *Distill*. AI Safety needs Social Scientists.

[6] **Ethan Perez**, Siddharth Karamcheti, Rob Fergus, Jason Weston, Douwe Kiela, Kyunghyun Cho. *EMNLP 2019*. Finding Generalizable Evidence by Learning to Convince Q&A Models.

[7] Angela Fan, Yacine Jernite*, **Ethan Perez***, David Grangier, Jason Weston, Michael Auli. *ACL 2019*. ELI5: Long-form Question Answering.

[8] Chenhao Tan, Vlad Niculae, Cristian Danescu-Niculescu-Mizil, Lillian Lee. *WWW 2016*. Winning Arguments: Interaction Dynamics and Persuasion Strategies in Good-faith Online Discussions.

[9] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, Percy Liang. *EMNLP 2016*. SQuAD: 100,000+ Questions for Machine Comprehension of Text.

[10] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, Christopher D. Manning. *EMNLP 2018*. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering.

[11] Paul Christiano, Buck Shlegeris, Dario Amodei. *arXiv 2018*. Supervising strong learnings by amplifying weak experts.

[12] Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, Marc'Aurelio Ranzato. *lample2018phrase-based*. Phrase-Based Neural Unsupervised Machine Translation.

[13] Guillaume Lample, Alexis Conneau. *NeurIPS 2019*. Cross-lingual Language Model Pretraining.

[14] Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, Matt Gardner. *NAACL 2019*. DROP: A Reading Comprehension Benchmark Requiring Discrete Reasoning Over Paragraphs.

[15] Divyansh Kaushik, Eduard Hovy, Zachary C. Lipton. *arXiv 2019*. Learning the Difference that Makes a Difference with Counterfactually-Augmented Data.

[16] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, Yejin Choi. *arXiv 2019*. WINOGRANDE: An Adversarial Winograd Schema Challenge at Scale.

[17] Rowan Zellers, Yonatan Bisk, Roy Schwartz, Yejin Choi. *EMNLP 2019*. SWAG: A Large-Scale Adversarial Dataset for Grounded Commonsense Inference.

[18] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, Yejin Choi. *ACL 2019*. HellaSwag: Can a Machine Really Finish Your Sentence?

[19] Rishabh Agarwal, Chen Liang, Dale Schuurmans, Mohammad Norouzi. *ICML 2019*. Learning to Generalize from Sparse and Underspecified Rewards.