My goal is to build language interfaces that can help people with real-world tasks: e.g., enabling a self-driving car to carry out "Get me to the airport," or allowing a digital assistant to respond to "My neck hurts." To build systems like these, standard natural language processing (NLP) techniques—which only map text to text, or map text to symbols—won't be enough. Our systems will need to disambiguate language in context (e.g., there may be several airports, which does the speaker mean?). Furthermore, systems will need to reason about goals (taking "my neck hurts" as a request for help) and predict interpretations (will a person safely understand medical instructions?). To take on these challenges, my work *grounds* language: tying language not only to perception and action, but also to people's goals and likely interpretations.

Language is often strategic. People choose what to say by predicting how listeners will interpret it and try to understand speakers by thinking about why they might have said what they did. Communication is a cooperative game, where speakers and listeners try to understand each other in order to succeed. This view, studied in linguistics as *pragmatics*, has helped me to build natural language systems that people can collaborate with. To generate language, existing NLP systems predict what to say—why not also predict how a listener will react? My work has shown that real-world language systems are more successful when they learn models of their human partners and use these models to reason about cooperative goals.

For language generation tasks like writing instructions, I've trained *listener* models to predict how people will interpret language and used these models to make generated language truthful and relevant [1]. For grounded understanding tasks, I've shown that *speaker* models of people's goals can help systems carry out instructions more accurately [2]. Constructing and training separate speaker and listener modules, which then reason about each other, makes them more likely to communicate successfully with people and do the right thing in context.

Beyond this modular speaker–listener framework, I've found that building and combining specialized modules is broadly useful in language grounding for complex tasks. A standard—and tempting—approach in deep learning is to build monolithic neural models that map inputs to outputs without taking advantage of any example-specific structure. Real-world grounding tasks, however, often need specialized computation: conditioning on diverse environmental contexts and carrying out a diverse set of subtasks. My work has developed neural models that use modules specialized to different decompositions of the task and different types of environmental context. I've shown that modular approaches often work better than their monolithic counterparts on complex grounding tasks like visual dialogue games with hidden context [3] and embodied instruction following [4, 5].

**Reasoning about interpretation**

Standard neural language generation systems are untruthful surprisingly often. For example, even state-of-the-art language models generate text that seems plausible but involves fake people or events, and document summarization systems often write summaries that don't follow from the document they're trying to summarize. Even when models are truthful, the language they generate may not be specific enough in context for a person to interpret correctly. For example, a system that is giving navigation directions may say "turn left" when there are two left turns available—a description that's technically true but inadequate in this particular context. My work has addressed these problems by showing that truthfulness and adequacy can be achieved by explicit pragmatic reasoning. To generate language pragmatically, we construct a model of how a human listener will interpret language and then have the generation system act optimally against this model listener (e.g., choose to say "make a slight left" so that the listener has a high chance of getting it correct).

My work on pragmatic generation has focused mostly on improving systems that generate grounded instructions: describing how to navigate through environments (e.g., the examples above) or manipulate objects (e.g., "Pour the red beaker into the blue one. Now add it to the green one on the right...") [1]. For each task, we construct and train a neural network listener model, typically a sequence-to-sequence model, which predicts how a person would carry out a given instruction in a world context. To generate instructions pragmatically, we search over instructions and choose one that has a high probability of making the listener

model carry out the correct sequence of actions.

Our approach improves substantially over standard baseline neural generation models. Baseline models struggle to learn to produce good descriptions from the available training data, so that people are usually unable to correctly follow the baselines' instructions. However, our pragmatic reasoning approach increases the percentage of times that people can correctly follow the instructions from 45% to 75%, so that our generated instructions are about as easy to follow as those written by other people. With collaborators, I've also shown that similar approaches improve state-of-the-art neural systems for data-to-text generation and document summarization [6].

**Reasoning about goals**

Reasoning about explicit models of human speakers also improves systems for the inverse task, grounded language interpretation. Language is often ambiguous, particularly in rich world contexts: a speaker may give the instruction "stop at the house" when there are multiple houses, some farther away than others. Reasoning about what the speaker chose not to say (e.g., "stop at the second house") can help resolve what goal they wanted the listener to achieve: stop at the first house, since the speaker would probably have described others differently. We've found that building and reasoning about a model of the human speaker provides another view on the training data that can provide substantial benefits.

Our pragmatic interpretation procedure is particularly successful on the task of vision-and-language navigation [2]. In this task, a system is placed in a StreetView-like interface showing panoramic images from the inside of a real building and needs to follow directions written by a person (e.g., "go down the hall, take a left, and stop at the exercise equipment"). Our approach uses a state-factored graph search to obtain candidate routes through the environment and chooses a route that has a high probability of making a learned speaker model produce the observed direction. We also showed that our speaker model can be used to synthetically augment the training data, by constructing new routes through the training environments and using the speaker to write directions for them. Our approach doubled performance over the state-of-the-art system at the time on a standard benchmark for this task, from 25% to 54% success. It's been exciting to see how other work on this task has built on our approach: e.g., using our speaker model in reinforcement learning training [7], improving the speaker and data augmentation [8], and extending our search procedure [9].

**Decomposing grounding with modules**

Language grounding has made incredible progress thanks to neural networks' ability to produce representations for text and images, combine these representations, and output decisions or actions. But these grounding models often fall short: they can ignore crucial parts of their inputs, or fail to generalize when they're evaluated on new combinations of things that were individually seen during training. The reason for this is that it's often easiest for a network to "cheat" and learn a relatively simple function that mostly solves a complex problem, but fails spectacularly when presented with something slightly different. My work has investigated *modularity* as a way to deal with this: decomposing a complex language grounding task into simpler parts, and learning and combining simpler neural network components which are specialized for each part.

In recent work, I've shown that modularity is an effective approach for visually-grounded dialogue tasks [3]. We built a system that can collaborate with people in a shared reference game: two partners, either people or agents, each see some part of an underlying board containing objects. The partners share some objects, but don't know which ones, and need to communicate and pool their information to pick one they both share. Our system uses a listener module to identify what a human partner might be talking about, a strategy module for choosing what to talk about next, and a speaker module to generate language. The listener and strategy modules are both neural conditional random fields (CRFs), which learn continuous representations of the visual world but also allow global reasoning about discrete relationships between objects. Our neural CRFs improve substantially over past unstructured neural models, which struggle at global

reasoning. We use the listener module to represent how a person might interpret the system's instructions, aiding the strategy and speaker modules in choosing both what to describe and how to describe it so that a human listener is likely to understand. We find that modularity and pragmatic reasoning both provide substantial improvements on the dialogue task. When paired with human partners, our system obtains a 50% relative improvement in success rate when compared to the past state-of-the-art neural dialogue system for this task.

In other work with collaborators, we've found that modularity also helps in embodied instruction following, where agents carry out tasks such as navigating around inside 3D virtual homes and interacting with objects. We've found benefits from two types of modularity: input modularity, where each module conditions on a specialized context, such as detected objects or the structure of the environment [4], and output modularity, where modules specialize in tasks such as navigating to locations or picking up objects [5]. In all settings, modules are trained jointly: the data forces each module to specialize its function but also to coordinate with the other modules to solve the task.

**Other work: Structured models for linguistic analysis**

In a separate line of work, I've designed search [10] and training [11] procedures for statistical syntactic parsing: constructing a tree-structured analysis of the syntax of a sentence. The parser we built produces accurate analyses of broad domain text [12], and the search procedure we developed for it has been used in recent work in computational psycholinguistics [13]. In work with collaborators, we've shown that syntactic analyses produced by our parser can be used to improve pre-trained neural models such as BERT, translating into downstream improvements to state-of-the-art models for a variety of core NLP tasks [14].

**Future directions**

As natural language becomes a standard interface to e.g., vehicles, appliances, and digital tutors, NLP as a field should pursue *interactive pragmatics*: building systems that model and learn from the people that the systems interact with in an iterated setting. Systems could benefit from adapting to people on two scales: on a long-term scale, adjusting to new domains and populations of users; but also on a short-term scale, building personalized common ground with individuals. Data is the bottleneck in these settings that require learning online or with people in the loop, posing difficulties for generic deep learning methods. Motivated by the successes of model-based methods at using data efficiently in other areas like reinforcement learning, I aim to build explicit models of the people that systems interact with. Since model-based methods are sensitive to how good the model is, we will want these models of people to be robust. In order to ensure the systems do what people want them to, we will also need the systems to infer and optimize for people's values. While fully modeling people's behavior poses a grand challenge, my work has taken steps toward modeling people's strategic *use of language*—and this has the potential to further support cooperative language-based interactions between people and machines.

A second direction is *broader grounding*: increasing the role of grounding in NLP. I'm excited to continue tying language to domains like vision and embodied control, e.g., for robotics. But I'm also excited about what a broader definition of grounding can do for more traditional NLP tasks. My past work has shown that modeling people's interpretations can improve some tasks like summarization that have typically been treated as only involving text. Large language models have now obtained unprecedented success in generating text that is fluent and plausible on the surface—but these models still hallucinate facts and are often misleading. It's now time to tackle challenges beyond fluency: for example, when a system generates an expository article, it should ground the article into facts about the world. A system that verifies articles might predict how a person's beliefs are likely to change from interpreting an article in context. Our language is directly informed by the richness of the world, and our language systems should be as well.

I'm excited to take further steps on problems like these in collaboration with others.

## References


[1] Daniel Fried, Jacob Andreas, and Dan Klein. "Unified Pragmatic Models for Generating and Following Instructions". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2018. URL: https://www.aclweb.org/anthology/N18-1177.

[2] Daniel Fried*, Ronghang Hu*, Volkan Cirik*, Anna Rohrbach, Jacob Andreas, Louis-Philippe Morency, Taylor Berg-Kirkpatrick, Kate Saenko, Dan Klein**, and Trevor Darrell**. "Speaker-Follower Models for Vision-and-Language Navigation". In: *Advances in Neural Information Processing Systems*. 2018. URL: https://proceedings.neurips.cc/paper/2018/file/6a81681a7af700c6385d36577ebec359-Paper.pdf.

[3] Daniel Fried, Justin Chiu, and Dan Klein. "Reference-Centric Models for Grounded Collaborative Dialogue". In: *in submission* (2020).

[4] Ronghang Hu, Daniel Fried, Anna Rohrbach, Dan Klein, Trevor Darrell, and Kate Saenko. "Are You Looking? Grounding to Multiple Modalities in Vision-and-Language Navigation". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2019. URL: https://www.aclweb.org/anthology/P19-1655.

[5] Rodolfo Corona, Daniel Fried, Coline Devin, Dan Klein, and Trevor Darrell. "Modular Networks for Compositional Instruction Following". In: *NAACL (to appear)*. 2021. URL: https://arxiv.org/abs/2010.12764.

[6] Sheng Shen, Daniel Fried, Jacob Andreas, and Dan Klein. "Pragmatically Informative Text Generation". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2019. URL: https://www.aclweb.org/anthology/N19-1410.

[7] Xin Wang, Qiuyuan Huang, Asli Celikyilmaz, Jianfeng Gao, Dinghan Shen, Yuan-Fang Wang, William Yang Wang, and Lei Zhang. "Reinforced Cross-Modal Matching and Self-Supervised Imitation Learning for Vision-Language Navigation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019.

[8] Hao Tan, Licheng Yu, and Mohit Bansal. "Learning to Navigate Unseen Environments: Back Translation with Environmental Dropout". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2019.

[9] Liyiming Ke, Xiujun Li, Yonatan Bisk, Ari Holtzman, Zhe Gan, Jingjing Liu, Jianfeng Gao, Yejin Choi, and Siddhartha Srinivasa. "Tactical Rewind: Self-Correction via Backtracking in Vision-And-Language Navigation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019.

[10] Daniel Fried*, Mitchell Stern*, and Dan Klein. "Improving Neural Parsing by Disentangling Model Combination and Reranking Effects". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*. July 2017. URL: https://www.aclweb.org/anthology/P17-2025.

[11] Daniel Fried and Dan Klein. "Policy Gradient as a Proxy for Dynamic Oracles in Constituency Parsing". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. 2018. URL: https://www.aclweb.org/anthology/P18-2075.

[12] Daniel Fried*, Nikita Kitaev*, and Dan Klein. "Cross-Domain Generalization of Neural Constituency Parsers". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2019. URL: https://www.aclweb.org/anthology/P19-1031.

[13] John Hale, Chris Dyer, Adhiguna Kuncoro, and Jonathan Brennan. "Finding syntax in human encephalography with beam search". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics)*. 2018. URL: https://www.aclweb.org/anthology/P18-1254/.

[14] Adhiguna Kuncoro*, Lingpeng Kong*, Daniel Fried*, Dani Yogatama, Laura Rimell, Chris Dyer, and Phil Blunsom. "Syntactic Structure Distillation Pretraining For Bidirectional Encoders". In: *Transactions of the Association for Computational Linguistics* (2020). URL: https://www.mitpressjournals.org/doi/full/10.1162/tacl_a_00345.