Research Statement

He He

Overview

The past several years have seen growing need for intelligent systems that work in real time and interact with humans, such as self-driving cars and personal assistance. Most existing learning methods are limited to a passive mode: the user feeds in some data and requests a prediction. This setting is often not sufficient for interactive systems. Consider the self-driving car as an example. It receives a stream of sensor data and must decide how to react to the ever-changing environment. At the same time, it needs to communicate with the passengers and respond to their requests. Interactive applications require us to rethink the learning paradigm. In a dynamic environment, the system no longer receives static, independent chunks of data to analyze. Instead, it makes a sequence of predictions given real-time data from different sources. Furthermore, user-centered applications require the system to actively exchange information with users and incorporate that information into its decision-making process.

My research is motivated by the insight that a model is always situated in a dynamic environment and that information acquisition is an integrative part of learning. Two themes of my work to date are: **learning in dynamic environments** and **interacting with humans**. I have developed efficient models that make dynamic predictions by deciding when and what information is needed for a particular instance, as well as agents that communicate and collaborate with humans. To develop effective interactive learning paradigms, one must combine innovations from machine learning, natural language processing, and human computer interaction. Therefore, my work spans from principled learning algorithms [1, 2] to live interactive platforms [3, 4].

Prior Work

Dynamic Information Acquisition for Efficient Prediction

An important distinction between a dynamic model and a static one is the capability of acquiring useful information on its own from various sources. Information is crucial to making predictions and it usually comes at some expense, ranging from computational cost such as hashing a string to monetary cost such as conducting a medical exam. Gathering all information is often not possible, nor necessary. For example, when doctors diagnose diseases, they start with the basic data (e.g. age, blood pressure, body temperature); only when these observations are not sufficient to make a decision do they move on to more costly tests such as MRI or X-ray. I formalized this problem as a sequential decision-making process: for each instance, the model uses a *policy* to sequentially select from a set of features based on past predictions until it is confident enough to finalize the prediction. The goal is to achieve a good trade-off between feature cost and prediction accuracy. Thus it learns which features are most cost-effective on an instance basis. I applied this framework to both multiclass classification [1] and structured prediction (dependency parsing [5]). The latter is one of the first few works that study feature computation cost in natural language processing, and our parser instantiates less than 30% feature templates without hurting the accuracy.

At a high level, our model learns to search a space—subset of features in the above setting. The initial success in feature selection motivated me to think about algorithms that learn to search the solution space of combinatorial optimization problems. Can we exploit common problem structures across instances to learn promising search paths? Can we predict the future cost of a node without expanding it? To this end, I designed an algorithm that learns to prune and prioritize nodes on a branch-and-bound tree for solving

Integer Linear Programming (ILP) problems. Given the same amount of budget (time or number of nodes explored), our solver consistently finds better solution compared to a commercial solver, Gurobi.

My approach is based on imitation learning: at training time, an *oracle* with knowledge of the ground truth can reverse-engineer the best features; thus we can learn a policy to mimic the oracle's behavior. The key challenge in this setting is that the oracle's policy may not be in our learning space. Therefore, I proposed an adaptive learning target that interpolates the oracle's decision and the learner's decision to provide a learning curriculum, which enjoys the no-regret guarantee and better empirical performance than directly learning from the oracle.

Incremental Language Understanding

The dynamic information acquisition setting assumes that all information is available *upon request*, which may not be true for real-time applications. Moving beyond this setting, I started to explore scenarios where the data comes in a stream. In the language domain, I built models that incrementally understand text, predict incoming words, and take actions given partial information [3, 6]. My particular focus in the area of incremental language processing is simultaneous interpretation. Unlike translating written text, the interpreter must translate *while* the speaker is talking. One challenge in this setting is that the critical information needed for a translation may not be available until some time in far future, making it hard to reduce translation delay without significantly hurting the translation quality. For example, when translating from verb-final languages (e.g. Japanese) to verb-medial languages (e.g. English), a naive translator often has to wait until the end for the verb to produce a sensible translation. Our findings [7] show that compared to translators, professional simultaneous interpreters use a unique set of skills to balance speed and accuracy, e.g. break long sentences into smaller, independent units, summarizing content, reordering words according to the source language grammar. Inspired by these observations, my work proposed the first framework [6] that unifies machine translation and interpretation techniques in a reinforcement learning setting, enabling later work on end-to-end neural interpretation systems [8, 9]. In addition to sentence segmentation which is widely used for simultaneous machine translation in prior work, we have shown that predicting future words [6] and reordering words [10] improve the speed-accuracy trade-off.

Interaction with Humans

Humans play a central role in many AI applications. In a user-centered application (e.g. personal assistance, smart home), the system need to interact with both the environment and the users to understand their needs. One desired component for such applications is a dialogue system, which enables users to communicate with the agent in natural language. To this end, my work [4] has studied a new symmetric collaborative dialogue setting, where two agents, each with private knowledge, must communicate to achieve a common goal, e.g. finding a common item in both agents' private knowledge bases. A typical dialogue in our setting consists of information exchange, question answering, coordination and inference. I designed a neural network model that bridges two main approaches to dialogue: semantic parsing that grounds utterances to logical forms and encoder-decoder models that embed utterances to a vector space. Our model combines structured knowledge and unstructured conversation context by connecting utterance embeddings to a knowledge graph. Human evaluation has shown that our dialogue agent can successfully complete the task by collaborating with users in a humanlike way.

Effective interaction requires an agent to infer about its partner's intention and adapt to their habits or preferences. To tackle this problem, I focused on an online game setting,[1] where players compete with each other to answer trivia questions as fast as possible. The goal is to design an agent that can play with humans and adapt to different playing strategies (e.g. cautious, aggressive). I proposed a deep reinforcement learning model which learns a latent distribution of player strategies as it plays with multiple users repeatedly [11]. The model then adjust its strategy depending on who it is playing against. We have shown that by explicitly modeling different user behavior, the agent is able to recognize different player's strategy and exploit weak players.

[1] `http://protobowl.com`

Future Work

As demonstrated in my prior work, I am interested in developing interactive approaches for real-time, user-centered applications. Moving toward the future, I plan to address new challenges arising from the interactive learning setting, such as robust interaction interface and goal-oriented natural language communication.

My ongoing work focuses on **robust dialogue systems**. One common failure mode of current data-driven dialogue models is that once the conversation deviates from the training data (i.e. the model no longer fully understands the dialogue context), the model falls to some disastrous state (e.g. repeating itself, generating generic responses). Such brittleness prevents us from deploying dialogue models and learning through user interaction. Currently, I am exploring two key ideas to tackle this problem. First, we often have strong prior knowledge about the user intention given a dialogue domain (e.g. booking tickets, chatting about movies), which allows us to build a high-precision parser that maps utterances to simple logical forms. Second, the dialogue agent has an operating zone where the user utterance can be parsed and responded to properly, and it should drive the conversation into this safe region. With these guidelines, our simple rule-based bot augmented with data-driven generation has achieved better performance on multiple tasks compared to more complex neural models. Looking forward, the end goal of this project is to build dialogue agents that improve themselves through chatting with people, much as modern search engines learn from user click feedback. This would require significant advancement in language generation, reinforcement learning, and human-in-the-loop learning, e.g. how to refine an utterance toward a certain goal, how to do safe exploration and solicit feedback in a conversation, and when to ask for help from a human to respond.

Modern intelligent agents need to interact with a population of users who have different preferences and habits. Expanding my work on opponent modeling [11], I am interested in developing **personalized, adaptive systems** that are automatically adjusted to different individuals and occasions. One example is *text style transfer*. People change their vocabulary and sentence structure depending on who they are addressing and where they are speaking, e.g. talking to their kid vs talking to their boss, tweeting vs writing an email. I have just started looking at how we can extract idiosyncratic patterns from text of a certain style and build models that apply these patterns to a given piece of text. When interacting with a population of users to learn strategies targeting at each individual, another challenge is to balance exploration and exploitation across multiple subjects. We do not want to repeatedly discover common knowledge about many individuals. One related application is *on-line education/tutoring systems*. While each student may prefer different amount of time on each topics, most of them would follow a common syllabus. To scale an interactive system to a large number of users with diverse needs, I will explore several research questions: How can we discover a shared "basis" of user behavior? How can we transfer knowledge among individuals and quickly adapt to new users or new scenarios? Is there an optimal strategy for selecting a human subject when exploring different actions?

The success of an interactive system hinges on the quality of feedback (reward signal) it receives. In many real world applications, the feedback is delayed and is insufficient to point the model in the right direction. Consider a dialogue agent troubleshooting a network failure with a user. The agent receives a 1–5 rating at the end of the conversation. The sparse reward, large space of natural language, plus non-stationary human users pose significant challenge to current learning methods—the model will need to interact with the world for a long time before it learns anything useful. To address these issues, I am interested in **eliciting and learning from rich supervision signals in language**. When humans communicate in natural language, oftentimes the feedback is implicit in their responses. For example, short, generic responses can indicate losing interest, such as "okay", "fair enough", and "I guess so". In more goal-oriented settings, they may also directly express negative feelings, e.g. "I don't want that.", "No, I was asking...". To take advantage of these signals, we need an empathetic component in the system that "senses" where things go wrong, thus alleviate the credit assignment problem. Another direction is to proactively ask users or a human teacher for stronger supervision. This can involve demonstration of a task, instruction in natural language, or paraphrasing a request. In this setting, we must also balance learning and the cost of richer supervision, which connects to my previous work on dynamic information acquisition. In addition to human feedback, we can extract prior knowledge from documents accompanying a task, such as manuals, guides, and instructions.

Overall, my goal is to develop robust, efficient interactive agents, with natural language as the key communication channel. I believe interaction is the key to the next-generation intelligent systems: It provides ways for a model to learn about the world it is situated in and better understand the user.

References

[1] He He, Hal Daum'e III, and Jason Eisner. Imitation learning by coaching. In *Neural Information Processing Systems (NIPS)*, 2012.

[2] He He, Hal Daum'e III, and Jason Eisner. Learning to search in branch and bound algorithms. In *Neural Information Processing Systems (NIPS)*, 2014.

[3] Jordan Boyd-Graber, Brianna Satinoff, He He, and Hal Daumé III. Besting the quiz master: Crowdsourcing incremental classification games. In *Empirical Methods in Natural Language Processing*, 2012.

[4] He He, Anusha Balakrishnan, Mihail Eric, and Percy Liang. Learning symmetric collaborative dialogue agents with dynamic knowledge graph embeddings. In *Association for Computational Linguistics (ACL)*, 2017.

[5] He He, Hal Daum'e III, and Jason Eisner. Dynamic feature selection for dependency parsing. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2013.

[6] Alvin C. Grissom II, He He, Jordan Boyd-Graber, John Morgan, and Hal Daumé III. Don't until the final verb wait: Reinforcement learning for simultaneous machine translation. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[7] He He, Jordan Boyd-Graber, and Hal Daum'e III. Interpretese vs. translationese: The uniqueness of human strategies in simultaneous interpretation. In *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2016.

[8] Jiatao Gu, Graham Neubig, Kyunghyun Cho, and Victor O. K. Li. Learning to translate in real-time with neural machine translation. In *EACL*, 2017.

[9] Khanh Nguyen, Hal Daumé, and Jordan L. Boyd-Graber. Reinforcement learning for bandit neural machine translation with simulated human feedback. In *EMNLP*, 2017.

[10] He He, Alvin Grissom II, Jordan Boyd-Graber, John Morgan, and Hal Daumé III. Syntax-based rewriting for simultaneous machine translation. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*, 2015.

[11] He He, Jordan Boyd-Graber, and Hal Daum'e III. Opponent modeling in deep reinforcement learning. In *Proceedings of the International Conference of Machine Learning*, 2016.