

留学報告書

安永迪弘

2021年6月

2019年9月より Stanford大学にてコンピュータサイエンス(CS)の博士学生をしております安永迪弘と申します。今回は2年目冬～春学期の生活を振り返りたいと思います。

1. 学会発表

前回のレポートで紹介した質問応答に関する研究が無事 North American Chapter of the Association for Computational Linguistics (NAACL) という自然言語処理の国際会議に採択され、6月に学会発表しました:

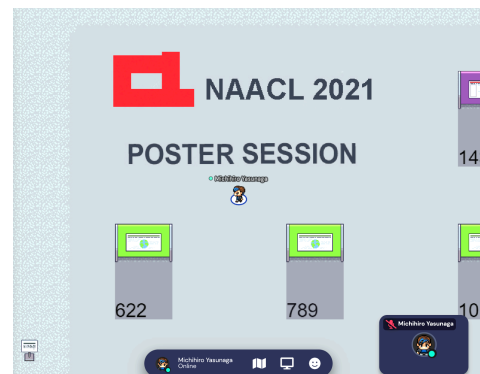
M. Yasunaga, H. Ren, A. Bosselut, P. Liang and J. Leskovec. "QA-GNN: Reasoning with Language Models and Knowledge Graphs for Question Answering" (<https://arxiv.org/abs/2104.06378>)

内容としては、質問応答システムに背景知識をどう効果的に取り込むことで精度と汎化性能を向上させるかという研究です。最近のAI研究では、背景知識の情報源として

- (1) 大量のテキストデータ(Wikipediaなど)で事前学習された言語モデル
- (2) 知識概念を構造化してまとめたデータベースである知識グラフ

の2つがよく使われていますが、両方を同時に使う方法論が十分に確立されていませんでした。そのため今回の論文では、その2つの情報源を1つの統合されたグラフに表現し質問応答システムに組み込むことで性能を上げる手法を提案しました。

引き続き学会はオンラインで、事前に15分のプレゼンを録画してアップロードし、当日はGather.Town (RPG風のバーチャル空間) でポスター発表をしました。不思議なことにバーチャルな学会にも慣れ、逆に便利で良いなとさえ感じるようになってきました。前大学での知り合いの多くとも再会し、ライフ・研究状況などを雑談できて楽しい時間になりました。



2. 研究

冬～春学期はdomain generalization (モデルの汎化) に関する研究をしました。機械学習モデルを実社会の問題に応用する際、学習時に使われたデータとは異なる状況 (学習データとテストデータのギャップ) に直面し、精度が大幅に落ちることがよくあります。例えば、いくつかの病院で学習された医療診断のモデルを別の病院で適用すると機能しなくなったり、事前学習されたプログラムのデバッグモデル¹ を新たなユーザーが書いたプログラムに適用すると精度を落としたりします。しかし人間の医師であれば基本どこの病院でも正しく診断できますし、経験のあるプログラマーであれば様々なプログラムをデバッグできるはずです。そのため、機械学習モデルが実際に使用される状況に汎化する方法を開発することが重要な研究になっています。

今回 domain generalization に関して2つのプロジェクトに取り組みました。1つは機械学習モデルを実社会の問題に応用する際に起こりうる、学習データとテストデータのギャップを想定した大規模ベンチマークの構築です。衛星画像や医療画像、遺伝子情報、化学化合物、SNS コメント、プログラムなど多岐にわたる応用分野において、それぞれの専門家から実際に起こりうるギャップを調査し現実に即したデータセットを作りました。[1] 2つ目は汎化の方法論に関する研究です。特にプログラムのデバッグの分野において、モデルが適用される domain (実際のユーザーが書いたプログラム) にあるラベルなしデータを利用することでモデルの汎化を大幅に向上させる手法を提案しました。[2]

ありがたいことに、この2つのプロジェクトは機械学習の国際学会 International Conference on Machine Learning (ICML) 2021に採択され、今年の夏発表の予定です:

[1] P. W. Koh*, S. Sagawa*, H. Marklund, S. M. Xie, M. Zhang, A. Balsubramani, W. Hu, M. Yasunaga, R. L. Phillips, S. Beery, J. Leskovec, A. Kundaje, E. Pierson, S. Levine, C. Finn, and P. Liang. "WILDS: A Benchmark of in-the-Wild Distribution Shifts" (<https://arxiv.org/abs/2012.07421>)

[2] M. Yasunaga and P. Liang. "Break-It-Fix-It: Unsupervised Learning for Program Repair" (<https://arxiv.org/abs/2106.06600>)

3. 授業

冬学期にCS 242: Programming Languages (プログラミング言語) の授業を取りました。Lambda Calculus などの原始的な言語を使って基礎となるプログラム (足し算, 割り算など)

¹ デバッグモデルの詳細については2020年6月の報告書を参照してみてください。

を書くところから始まり、その後徐々に高級言語にある機能 (typeシステム, プログラムの continuation) を実装していきました。一言で言えば、プログラミング言語がどう発展してきたのか勉強でき、コンピューターサイエンスを専門にする者として履修しておいて良かったと感じました。CS PhD学生は卒業までに理論・システム (プログラミング言語やコンパイラー)・応用 (AIなど) の3分野で各2つずつ授業を履修することが必要ですが、今回でこのrequirementを完了することができました。

4. 最後に

これまでのPhD 2年間で振り返ると、二人の指導教授の下、面白いと思った様々な研究課題に取り組み、ラボメイトとの共同研究の機会も増え、研究と知識の幅が広がったと実感しています。と同時に、指導教授には「深さ」(1つのトピックを究めて認知されること) がまだ必要だと言われており、頑張っていこうと思っています。船井財団にはいつもサポートしていただき、本当に感謝しております。