Neural NLP Deep Dive

Mengyu Jackson

Last week, I read a summary paper detailing advancements in NLP up to 2019, and I learned a lot of very exciting progress has been made since I last focused on NLP. In such a fast-moving field, a lot can happen in a few years, so today I'm going to share some take-aways from [another paper]() from June of 2021, as well as a few things from a recent (Dec 2021) [blog post]() by Amal Menzli. Two of the most exciting advancements in NLP that I read about last week were 1.) Use of Neural Networks (especially deep learning) and 2.) Graph-based algorithms. This week's paper (Graph Neural Networks for Natural Language Processing: A Survey) covers both of these topics in great detail. To follow along and help gain practical experience, I read the blog post to decide on a Python library for Graph Neural Networks and got course recommendations as a bonus. I decided to avoid Graph Nets due to the dependency on Tensorflow 1, and check out each of the other three recommended projects since they all looked promising (which I'll review in a future blog post).

Back to the research paper, there are 87 pages explaining the current state of the art for Graph Neural Network techniques, followed by 40 full pages of references for further reading. Although the algorithms outlined in sections 4-6 are very interesting and I will certainly refer back to them frequently as I continue on my personal NLP projects, I found section 7 (Applications) to be the most exciting chapter on my first reading, and the most important section for anyone who wants to read more than the Abstract and Summary but can't finish the entire paper. Table 3 (partially reproduced below) is amazing, breaking down the NLP tasks by application and evaluation metrics, and listing the most recent and relevant papers for each task.

Table 3: Typical NLP applications and relevant works using GNNs

Application	Task	Evaluation	References
NLG	Neural Machine Translation	BLEU	Bastings et al. (2017); Beck et al. (2018); Cai and Lam (2020c) Guo et al. (2019c); Marcheggiani et al. (2018); Shaw et al. (2018) Song et al. (2019); Xiao et al. (2019); Xu et al. (2020c); Yin et al. (2020)
	Summarization	ROUGE	Xu et al. (2020a); Wang et al. (2019e); Li et al. (2020b) Fernandes et al. (2019); Wang et al. (2020a) Cui et al. (2020b); Jia et al. (2020); Zhao et al. (2020a) Jin et al. (2020b); Yasunaga et al. (2017); LeClair et al. (2020)
	Structural-data to Text	BLEU, METEOR	Bai et al. (2020); Jin and Gildea (2020); Xu et al. (2018a) Beck et al. (2018); Cai and Lam (2020b); Zhu et al. (2019c) Cai and Lam (2020c); Ribeiro et al. (2019b); Song et al. (2020) Wang et al. (2020f); Yao et al. (2018); Zhang et al. (2020d)
	Natural Question Generation	BLEU, METEOR, ROUGE	Chen et al. (2020g); Liu et al. (2019b); Pan et al. (2020) Wang et al. (2020d); Sachan et al. (2020); Su et al. (2020)
MRC and QA	Machine Reading Comprehension	F1, Exact Match	De Cao et al. (2018); Cao et al. (2019b); Chen et al. (2020d) Qiu et al. (2019); Schlichtkrull et al. (2018); Tang et al. (2020c) Tu et al. (2019b); Song et al. (2018b) Fang et al. (2020b); Zheng and Kordjamshidi (2020)
	Knowledge Base Question Answering	F1, Accuracy	Feng et al. (2020b); Sorokin and Gurevych (2018b) Santoro et al. (2017); Yasunaga et al. (2021)
	Open-domain Question Answering	Hits@1, F1	Han et al. (2020); Sun et al. (2019b, 2018a)

Although all of the applications and tasks have had some success with GNNs in recent years, the application that interested me the most was the MRC and QA application. The Machine Reading Comprehension and Question Answering tasks seem extremely practical and valuable, and the graph-based representation provides many advantages over seq2seq models.

Graph-based approaches can handle distant relationships between topics or words better because edges can be directly introduced between elements that are distant in the input sequence. Many types of conversations that could automatically be handled using a knowledge base and an NLP engine benefit from this.

When troubleshooting technical issues, for example, it is very common to try multiple approaches to solve the original problem (“Try different web browser” “Try logging out and back in” “Try using the app” etc.). Each troubleshooting attempt has a strong connection to the original problem, but there’s not a meaningful relationship between different troubleshooting attempts. A graph-based approach seems like a better fit for a conversation style that continuously circles back to a central topic or theme introduced when the conversation started while frequently forgetting more recent parts of the conversation.

Now that I have a more focused problem statement (“I want to work on Machine Reading Comprehension/Question Answering”), a clear idea of which Python libraries I want to use to solve the problem, and a wonderful set of references for further reading when I get stuck, I’m ready to dive in and start writing code. I’ll share my experiences with implementing a GNN based NLP engine in my next update on this topic. Happy Holidays!