A Neural Knowledge Engine for Rapid Generalization on QA Tasks

Introduction

Machine reading, or basic question-answering, is the task of understanding a textual passage. You might have a few sentences in the passage, and the model would have to answer a question about the passage, to show that it “understood” what was going on. How do you build models. Question-answering is a crucial task in NLP, and numerous high-quality datasets like SQuAD and NewsQA have come about in recent years.

In question-answering, the task is to answer the question given direct reference to the passage. However, one might want to extend that model to situations where you read the passage before, and answer the question after. In an extreme case, you might imagine reading the entirety of Wikipedia, and then answering a question about it afterwards. In that scenario, a reasonable thing to do would be to suppose that the machine has access to some form of knowledge representation that can be efficiently queried.

What kind of representation will that be? Surely, it can’t be the concatenated BiLSTM output of the entirety of Wikipedia. Distributed representations of words have shown to provide useful features for many tasks in NLP (Peters et al. 2018). Methods like GLoVe and word2vec are rather successful in capturing large-scale similarities between words in a meaningful way. However, encoding more complex things, like sentences, is still an open problem in the NLP community. Learning good representations of structured knowledge, whether in a supervised or unsupervised way, will likely mean major improvements in tasks ranging from logical entailment to open-ended dialogue to question-answering, machine reading, and information retrieval.

Unsupervised learning is surprisingly effective at capturing meaningful features of language (Radford, Jozefowicz, and Sutskever 2017). Techniques like auto-encoding and pair-similarity have yielded meaningful signals in domains like speech, vision, and text. (Conneau et al. 2017) show that supervised learning of sentence representations based on the Stanford Natural Language Inference dataset, one of the highest-quality resources constructed for understanding sentence semantics, outperform other methods like SkipThought.

However, the verdict is still out as to whether one can recover the same types of representations in a more organized way: what can we do to encourage the model to learn useful features like sentiment? Can we design more useful inductive biases, like convolutions or residual connections, or can we design tasks like multi-lingual translation that encourage useful, general representations?

If we think about question answering as a means to more effective reading comprehension, one way to separate the task is to consider the various components of answering questions. We propose that QA systems can be further decomposed into two stages: “reading”, going from text
to representation, and “reasoning”, performing operations on that representation to get to the right state.

In this project, we’ll focus on generalizing graph representations. (Battaglia et al. 2018) provide a huge survey on the current state and future directions of graph-based reasoning networks, which extend, unify, and generalize a lot of recent models that involve entity-based reasoning. For visual question-answering and regular question-answering both, dealing with entities have helped many methods perform better on a lot of tasks.

**Related Work**

In (He et al. 2017), a task-oriented dialogue system is described, where there are two symmetric agents A and B, who have to remember the state of the dialogue in order to figure out which object they have in common. The way the knowledge is represented in a relational way, similar to EntNet, is useful.

**Problem**

Formally, we want to use question-answering as an auxiliary task for building knowledge representations.

**Timeline**

**Sep 20** – Build model, run baselines
- Run EntNet on SQuAD (Henaff et al. 2016)
- Build the model and run it on SQuAD

**Oct 20** – start evaluations/experiments
- Run another comparison model on reading comprehension
- Experiment with different settings, etc
- Build tasks to test specific capabilities

**Nov 20** – first draft of paper
- Continue evaluating model
- Get feedback on paper

**Dec 10** – submit paper to NAACL 2019