CS224N Project Ideas

Instructions for CS224n students:
If you are interested in doing one of these projects, get in touch with the associated mentor to arrange a meeting. We encourage you to do this early!

Title: Towards better character-based word vectors
Contact: Peng Qi (penggi@cs.stanford.edu)
Description: Character-based word vectors like FastText implicitly assume that the same morpheme is equally "important" regardless of the context, but in reality that's not the case. In this project, we'll explore effective ways to incorporate this intuition to improve the quality of these word vectors.

Title: Multi-hop question answering in the wild
Contact: Peng Qi (penggi@cs.stanford.edu)
Description: Most existing question answering systems published work under the assumption that all evidence necessary for answering the question can be found in one local context (e.g., one Wikipedia document). This precludes them from answering questions like "When was 2018's highest grossing movie released?" In this project, we will explore relatively simple but effective methods to address these questions in the context of HotpotQA, a recently released question answering dataset for multi-hop reasoning in the wild.
Comments: Groups of more than one student preferred, especially if they had experience with research. This project might be slightly more engineering-heavy than others.

Title: Faster Transformers
Contact: Kevin Clark (kevclark@cs.stanford.edu)
Description: A recently proposed neural network architecture called the Transformer works very well for many NLP tasks. However, its runtime is quadratic in the length of the input sequence, which means it can be slow when processing long documents or taking characters (rather than words) as inputs. In this project, you would explore ways of modifying the Transformer architecture so it runs much faster while sacrificing as little accuracy as possible.

Title: Improving NLP Representation Learning with Definitions
Contact: Andrey Kurenkov (andreyk@stanford.edu)
Description: The recent paper "Auto-Encoding Dictionary Definitions into Consistent Word Embeddings" showed that word vectors can be significantly improved by training on the task of dictionary definition auto-encoding. At the same time, word vectors are increasingly becoming less relevant as full model pretraining is becoming the norm with models such as ELMo and ULMFiT. But, what task should pretraining done with? "Looking for ELMo's friends: Sentence-Level Pretraining Beyond Language Modeling" showed that using multiple tasks is a useful idea for representation learning. This project's main aim is to explore the use of definitions for such representation learning.

Title: Automated Glossary Construction
Contact: Vinay K. Chaudhri (vinayc@stanford.edu)
Description: We are constructing an intelligent textbook in Biology that contains an explicit coding of the knowledge in the book. See http://web.stanford.edu/~vinayc/intelligent-life/ for more details. At present, the coding of the knowledge is all done manually which is expensive and does not scale. We are interested in seeing which aspects of the coding could be automated using NLP. We made a start toward this goal in a project http://web.stanford.edu/~vinayc/intelligent-life/Fall-2018-CS229-Project.pdf We are interested to move this work forward to improve the accuracy and coverage.

Title: Analogical Reasoning
Contact: Vinay K. Chaudhri (vinayc@stanford.edu)
Description: A popular test for checking if a program or a computer understands concepts is to ask questions of the form A is to B as C is to D? This has been posed as Tasl2 on SemEval2012. See https://www.cs.york.ac.uk/semeval-2012/task2.html There have been several approaches based on neural embeddings that have done well on this task. For example, see: https://www.aclweb.org/anthology/N13-1090 and https://levyomer.wordpress.com/2014/04/25/linguistic-regularities-in-sparse-and-explicit-word-representations/ We are interested in evaluating this approach in the context of a Biology textbook.

Title: Using Determinantal point processes for improving beam search samples
Contact: Ashwin Paranjape (ashwinp@cs.stanford.edu)
Description: Many language generation tasks employ beam search over the outputs of a neural decoder to obtain a high probability output. However, the samples at each step are generated using independent samples based on the probability distribution of the output tokens. This leads to the beam getting populated with correlated samples reducing the efficacy of the beam. Determinantal point processes (DPP, https://arxiv.org/abs/1207.6083) can be used to obtain diverse samples from a probability distribution with the similarity of samples encoded using the output embeddings. This concrete steps involved in the project are quantifying the effect of correlated samples, studying the applicability and computational feasibility of DPPs for diverse sampling and then applying DPP to NLG tasks to show improvement in beam search generation in real world tasks.

Title: Compositional pre-training for semantic parsing
Contact: Robin Jia (robinjia@stanford.edu)
Description: Recent success stories like BERT have shown the effectiveness of pre-training a large model on unlabeled text, then fine-tuning it for the desired end task. In this project, we will try to extend this idea to tasks that involve both natural language and formal language (e.g., SQL queries). We will focus on semantic parsing, in which a model receives a natural language utterance (e.g. “How many states border Illinois?”) and outputs an executable logical form (e.g., a database query that finds the states in a “borders” relation with “Illinois”, and returns the size of this set). While it is possible to directly train a neural sequence-to-sequence model to do this task, such models often fail to learn the right *compositional* structure: sub-formulas within the logical form should correspond roughly to

syntactic constituents in the utterance (e.g. “How many” translates to a “count” operation, and “border Illinois” independently translates to a relational query). Can we encourage neural networks to learn this structure by pre-training them on compositionally-generated data? Possibilities include synthesizing pseudo-natural utterances called “canonical forms”, or using a technique called data recombination to augment an initial dataset.