CS 4650/7650: Natural Language Processing

Project

Diyi Yang

Announcements

- Homework 1
- Homework 2 due: Feb 3rd, 3:00pm ET

Midterm (In Class)

- One-page cheat sheet (A4 size)
- Feb 26th

Project Logistics

- Team info: March 4th
- Proposal: March 11th
- Project proposal feedback meetings: March 23rd
- Midway report: Apr 1st
- Final presentations: ~ Apr 20th
- Final report due: ~ Apr 23rd

Pick Your Project

- Clearly define a specific goal/hypothesis of the project
  - e.g., propose a new NLP task or a new method, or reimplement a classical paper
- Pick an achievable goal (we can help!)

Below slides: credits to UC Berkeley and Princeton NLP projects guidelines

Design Your Project

- Availability of data
  - Not recommend to collect your own data
- ML framework
  - sklearn, keras, pytorch, Tensorflow
- Statistical models or neural network architecture
- Availability of computation

Literature Survey

- Do a thorough literature search
  - Google scholar, ACL anthology
- Search “awesome {NLP, RL, computer vision} papers github”
  - Example: https://github.com/mhagiwara/100-nlp-papers
  - Play around with code existing on github and see how readable/usable it is
  - https://paperswithcode.com/sota

Tips for Reading Papers

1. Do not need to read from the beginning to the end in order
2. Tables, figures, captions provide useful information at first glance
3. Plenty of blogs, github repos, etc. that summarize several papers at once in a nice manner

Types of Projects

1. Experiment with improving an architecture on a well defined NLP task
2. Case study: Apply an architecture to a dataset in the real world (that has not been done before)
3. Compete in a predefined competition (SemEval 2020, Kaggle, etc.)
4. Stress test or comparison study of known models/architecture (e.g. when are RNNs better than Transformers for task XYZ?)
5. Design a novel NN layer, objective function, optimizer, etc.
6. Multi-domain NLP (RL + NLP, CV + NLP, Social Science + NLP …)
7. Visualization/Interpretability study of deep learning models
8. …

Resources

- Your own/group/advisor's resources
- Google Cloud/Amazon AWS credits/Google Colab (1 free GPU)
- Request/get access to the above ASAP if you plan on using them!

The Dos (Tips for Successful Projects)

1. Clearly divide work between team members for optimal progress
2. Start early and work on it every week rather than rush at the end
3. Set up work flow - download data, verify data, set up base code
4. Have a clear, well-defined hypothesis to be tested (++ novel/creative hypothesis)
5. Conclusions and results should teach the reader something
6. Meaningful tables, plots to display the key results
   - ++ nice visualizations or interactive demos
   - ++ novel/impressive engineering feat
   - ++ good results

The Don’ts

1. Data not available or hard to get access to, which stalls progress
2. All experiments run with prepackaged source - no extra code written for model/data processing
3. Team starts late - only draft of code up before dues
4. Just ran model once or twice on the data and reported results (not much hyperparameter search)
5. A few standard graphs: loss curves, accuracy, without any analysis
6. Results/Conclusion don’t say much besides that it didn’t work
   - Even if results are negative, analyze them

This Stage

1. **Find a team**
   - Team info: March 4th, 2020
   - Signup link: **https://forms.gle/jmWJsgALun2aLdik9**
2. **Brainstorming**
   - Have each team member come up with ideas
   - Refine & filter out ideas
     - Data availability
     - Has the same idea been done before (with possibly existing github code)?
     - How long and how much do the models need to be trained?