CS378 Final Project

Proposal Due Date: Friday April 12 at 5:00pm CST
Final Report Due Date: Tuesday, May 14 at 5:00pm CST (hard deadline)

Collaboration You are free to work on this project in teams of two (strongly encouraged) or individually. Individual projects can be less ambitious but should not be less complete: a half-implemented system does not make a good project outcome. All partners should contribute equally to the submission, and all partners will receive the same grade for it. You may collaborate with a person from outside the course as well in case you're also using this final project for another course. You are also free to discuss your project with others in the course, though only the people on your team should contribute to the actual implementation/experimentation involved. Any external resources used must be clearly cited.

Assignment

You have three options for your final project (or infinitely many options, depending on how you view it):

1. Experiment with and extend a machine translation system
2. Implement a nontrivial Alexa skill
3. Something else you want to do

These are all significantly more open-ended than past assignments in this class, hence why you are given additional time to tackle them. You will submit a proposal to the course staff, we will provide feedback, and then you will execute the project over the last weeks of the class.

Machine Translation

One final project option is to experiment with and extend a machine translation system as discussed in lecture. You can either choose to operate in the phrase-based or neural paradigm. Each has its own challenges.

Roughly, your project should follow this template:

1. Identify training data. You should pick a language pair to start with. You may find it easier to work with languages that you know. If English is the language you know best, translating into English may be easier to wrap your head around than translating into other languages; you'll be able to easily diagnose errors in English and use standard English resources on the target side.
2. Get a baseline model working for your approach.
3. Try some sort of extension to that model or do some kind of open-ended investigation. Your proposal should describe this in as much detail as you can. Finding recent papers in the literature and seeing if they have interesting ideas can be a good starting point. You are allowed to reimplement something that's shown up previously, **as long as the implementation is your own**.

Tools In either case, you may wish to start with an existing implementation or toolkit so you can start to make progress without building a large baseline system. For neural machine translation, OpenNMT[1] is the best place to start. For phrase-based, Moses[2] is the most standard framework.

[1] `http://opennmt.net/`
[2] `http://www.statmt.org/moses/`

Data There are several common corpora that are publicly available for machine translation. A good place to start is the WMT machine translation tasks.[3] This features data drawn from Europarl[4] and the UN corpus.[5] Any of these corpora can work.

Compute and Feasibility **Large MT datasets can be very large and these methods can be slow!** You probably want to do something like restrict the vocabulary and sentence length to begin with, then scale up from there once you have something working. However, unless you have access to a GPU of your own, you should probably just not plan on doing many training runs of large neural MT systems. We will try to make sure your proposal stays feasible, but this should be something you think about.

Scope What you try as an extension does not strictly need to work. However, if it doesn't work, you should have a plan for how to analyze it and be able to dig into the data more. Just saying "I tried X and got it 90% built but it still crashes" is not a good project outcome.

Dialog System

The second suggested project is to build some kind of conversational dialog system. Roughly, this might look like:

1. Get a basic Alexa skill working.
2. Extend that Alexa skill in some nontrivial way. This might involve additional language understanding (beyond what's possible), better generation, or sophisticated dialogue state management (see below).

The course staff are less familiar with Alexa in general. However, we do have access to some support directly from Amazon. Interested teams can contact the course staff with questions and we can put you in touch with people at Amazon.

Resources EE596 at the University of Washington[6] deals extensively with creating Amazon Alexa skills. In particular, Lab 1 and Lab 2 describe how to create an account on AWS, get a simple skill working, and start customizing it more deeply.

The course staff has have ten Amazon Echo Dot devices for use as well as three Amazon Echo Shows. You can develop and debug on Alexa in AWS without these devices, but of course you may want to use them for testing, showing off what you've done, etc. Depending on demand, we may be able to issue each team their own device (to be returned at the end of the semester), or we may need to devise a policy for sharing these.

Scope Building a basic Alexa skill is *not* a suitable final project. What you do should go significantly beyond what's possible based on following a simple walkthrough for developing an Alexa skill.

One avenue you might consider is trying to do more sophisticated language understanding than is possible with the current intents and slots model, which might look like having a custom slot and then your own code that handles it in a sophisticated way.[7] Alternatively, you could explore modeling dialogue state in a more complex way.

[3] `http://statmt.org/wmt17/translation-task.html`
[4] `http://www.statmt.org/europarl/`
[5] `http://opus.nlpl.eu/UN.php`
[6] `https://hao-fang.github.io/ee596_spr2018/syllabus.html`
[7] See `https://developer.amazon.com/docs/custom-skills/slot-type-reference.html#phrase-types`

Your own project

Finally, you can pursue your own independently-chosen final project. Your proposal in this case should, as precisely as possible, describe the problem you want to solve or understand and the work you propose to do to solve it. Make sure you describe any datasets you plan to use and demonstrate that what you're proposing is feasible. Working on adjacent areas of machine learning (e.g., computer vision or robotics) is okay as long as there is a reasonable connection to concepts from this course. Ask the course staff if you're unsure.

Scope The expected scope should be similar to the MT and dialog projects. In particular, you should either have a nontrivial piece of implementation (putting together existing pieces can be nontrivial) or analysis.

Deliverables

The final project is worth 30% of your course grade. The deliverables are as follows.

Proposal You should turn in a $\frac{1}{3}$ **to 1 page proposal** on the proposal due date. This proposal should outline what you propose to do and a rough plan for how you will pursue the project (e.g., "we propose to download X system, run it, then implement our system on top of their framework and compare the results"). The course staff will then provide feedback and guidance on the direction to maximize the project's change of succeeding. **The proposal is not graded.**

Final Report The primary deliverable is a paper written in the style of an ACL[8]/NeurIPS/etc. conference submission.[9] It should begin with an abstract and introduction, clearly describe the proposed idea or exploration, present technical details, give results, compare to baselines, provide analysis and discussion of the results, and cite any sources you used.

This paper should be between 4 and 8 pages excluding references. Different projects may take different numbers of pages to describe. If you have lots of analysis and discussion or are trying something more ambitious, your paper might be longer; if you're implementing something complex but succinctly described, your paper might be shorter.

Your project is *not* graded solely on the basis of results. You should approach the work in such a way that success isn't all-or-nothing. You should be able to show results, describe some successes, and analyze why things worked or didn't work beyond "my code errored out." Think about structuring your work in a few phases so that even if everything you set out to do isn't successful, you've at least gotten something working, run some experiments, and gotten some kind of results to report.

Grading: We will grade the projects according to the following rubric:

- **Scope/Plausibility (35 points)**: Is the idea of sufficient depth for a course project? Does it follow the guidelines suggested above?
- **Implementation/Soundness (25 points)**: Is the idea technically sound? Do you describe what seems like a convincing implementation? Is the experimental design correct?

for information about phrase slots, which can be used to get free text input out of Alexa (as opposed to highly restrictive slot values). More here: `https://developer.amazon.com/blogs/alexa/post/a2716002-0f50-4587-b038-31ce631c0c07/enhance-speech-recognition-of-your-alexa-skills-with-phrase-slots-and-amazon-searchquery`

[8] Style files available here: `http://www.acl2019.org/EN/call-for-papers.xhtml`

[9] The Iyyer et al. paper is a good example of this: `https://people.cs.umass.edu/~miyyer/pubs/2015_acl_dan.pdf`

- **Results/Analysis (25 points)** Whether the results are positive or negative, try to motivate them by providing examples and analysis. If things worked, what error classes are reduced? If things didn't work, why might that be? What aspects of the data/model might not be right? You should try to report results for a baseline from the literature, your own baseline, your best model, and possibly results of ablation experiments.
- **Clarity/Writing (10 points)**: Your paper should clearly convey a core idea/hypothesis, describe how you tested it/what you built, and situate it with respect to related work as best you can.[10]

Presentation There is no mandated final project presentation. However, some of you may have neat projects that you want to show off to your peers. Time will be set aside on the last class day for any teams that are interested to give a brief presentation/demo of what they've done. If there's sufficient interest, we may take up some of the second-to-last day as well.

Computational Resources Available

This course has an allocation on TACC. Depending on demand, each group can likely have around 10 hours, which is not a lot. Try to reserve this for when your model is working and you need to run full-scale experiments.

If you haven't already used them for other courses, Google Cloud provides free credits when you sign up, which may be useful as well.

Submission

You should submit your final report in a single PDF on Canvas. No other datasets, code, results, etc. need to be uploaded. **Late projects will not be accepted.**

[10] Some advice here: `http://www.cs.utexas.edu/~gdurrett/courses/fa2017/lectures/lec8-writing-4pp.pdf`