HW4 Discussion

Jessica Sweeney

February 18, 2021

Time Performance

The overall time investment for the rule-based system was lower than for the statistical system. In our case, this is because we had already implemented an outline of an NLU system in the previous homework, and the domain we're working with is fairly constrained. The statistical system had a higher up-front cost because we had to understand and debug the models, and perform some processing on the input data to convert it into a trainable format (i.e. converting the HTML tags into a BIO tagging scheme for the CRF). However, now that we have both models set up, the statistical model would be far easier and faster to use in the future for this task if we suddenly needed to handle new tags or change domains. As long as we had annotated data for the new task, we could retrain the model without any extra work. Extending the rule-based system, however, would take some time to understand the scope of changes we needed to make, and how they interact with our existing rules.

Another time consideration was that since the rule-based system doesn't require any training, it's very quick to get up and running, but the statistical system takes about 30 seconds to retrain every time we made updates.

Fixing Errors

Fixing errors made by the models was very straightforward in the rule-based case, since we could follow the chain of logic in our code, identify where a regex was matching something we didn't want it to, or where our intents were being overwritten by later regexes, and make small, localized changes. Fixing the statistical errors was somewhat more difficult, and also less effective. Firstly, it was hard to identify why an error was being made: not enough training data? lack of features? And secondly, it was hard to determine what changes to the model would fix it. For example, we noticed that our statistical model wasn't properly tagging names, so we added some lexicon containment features that covered the names we saw in the test set that weren't being tagged. However, any new names in deployment wouldn't have this feature, and names that weren't in training but were in the test set would also not have had enough training signal for the model to pick up on them. Making this fix only raised our accuracy very slightly, and didn't cover all the cases. Handling names more robustly would require a lot more training data, or a larger lexicon of names, or a more powerful learning model.

ASR

We didn't make any ASR updates to the statistical system in Q4, because the problem seemed to us to be mostly a problem with the training data. Since our model wasn't trained on data with ASR gaps, it sometimes struggled with large gaps in context. Some of the errors it was making were also unrelated to the ASR errors, and were just a result of the system being imperfect even with full context.

We didn't update the rule-based model to recognize and handle the ASR gaps, but we did add a few extra words to the regexes for intents like REORDER, to capture some cases where a crucial word we were already checking for was gapped out, but there was another word in the order that identified it as REORDER that we should also check for.

Other issues

Data labelling was the most time-consuming part of this assignment. We tried to do data processing on the other groups' annotations in order to map them all to a common tagging system, but the number of errors we realized we would have to handle (typos, inconsistencies, tags that were in the data but not the provided mapping, missing start or end HTML tags) made it completely unrealistic to pull it all together, and in the end we used just our data. Needing to label the test datasets also took a lot of time.

One other issue we ran into during the "improvements" questions was not implementing changes that would overfit to the data we were evaluating on. It was tempting to handle things case-by-case, or add phrases to our regexes that we knew would break if we got new data. If we had been implementing this system for a real-world application, we would have needed to think harder about whether we wanted to make certain fixes just for the sake of higher eval numbers, or in order to make the system actually function better.