Hierarchical Generalisation without Hierarchical Bias in RNNs

Course project for Computational Linguistics I.

Team:
- Saujas Srinivasa Vaduguru (20171098)
- Ujwal Narayan (20171170)

Project Description

Our project aims to study how recurennt neural networks perform on tasks that require hierarchial generalisation. We study the specific case of transforming declarative sentences to polar questions in English. We generate the declrative sentences using context free grammars and transform them using a deterministic rule to generate the data. Then, we train seq2seq models on the genrated data. We evaluate the generated data on a test set with sentences similar to the ones presented during training and a generalisation set that requires the model to learn the correct the heirarchial rule for high accuracy. We perform the same task for multiple models and observe the effects of different random initialisations on the network.

Literature Review

Our project is a study of McCoy et al (2018) which studies the same problem and goes deeper into analysis. The McCoy et al paper is inspired by the work of Frank and Mathis (2007), which uses recurrent neural networks to transform declarative sentences into questions.

McCoy et al differs from Frank and Mathis in the use of seq2seq networks as described in Botvinick and Plaut (2006) and Sutskever, Vinyals and Le (2014).
McCoy et al also briefly notes comparisons between the performance of their model and mistakes made by human children in acquiring the same rule.

Methods

We generate the sentences used as data using context free grammars. For each of the different kinds of sentences that are present in the data, we have a different context free grammar. If this was not the case, we would have to twice the work to parse the sentences again, and apply the transformation rule to get the question. We use non-recursive CFGs, but use only a small fraction of the total number of sentences that can possible be generated using the CFG.

We also use two different languages -- no-agreement and agreement language -- which differ in the auxiliaries. This will allow us to check for variation in performance with different syntactic cues, since the auxliaries like do and don't agree with the number of the noun, while auxliaries like can and will do not change with the number of the noun.
In the context free grammar we use fixed indices to indicate the position of the main auxilliary which was removed before the final processing, and used to carry out the transformation from declarative sentences to questions in a deterministic way.

To ensure that we have sufficient variety and that the generation algorithm runs to completion in time, we sample random parts of the vocabulary (while being sensitive to parts-of-speech).

We use three different data sets -- the training set, the test set, and the generalisation set. The generalisation set contains examples that are held out of the training set, and are used to determine whether the model learns the hierarchical rule.

The (approximate) statistics for the different datasets are noted in the code. The overall stats for the sets are:

Set	No-agreement	Agreement
Train	118k	117k
Test	9k	9k
Generalisation	10k	10k

We then train multiple seq2seq models (written using the PyTorch deep learning framework) on the data, and evaluate it against the test and generalisation sets. The seq2seq models use the GRU gating mechanism, augmented with the attention mechanism.

Experiments

Experimented with 4 different random initialisations of the hidden layers. For each initialisation, statistics for performance on test and generalisation data were collected.

Test accuracy

No-agreement

Model #	Word Match Accuracy	POS Match Accuracy
1	84.5	93.2
2	86.9	94.7
3	91.3	97.5
4	84.7	92.7

Agreement

Model #	Word Match Accuracy	POS Match Accuracy
1	87.0	89.6
2	95.4	98.7
3	98.8	99.9
4	83.8	87.4

Generalisation Accuracy

No-agreement

Model #	Word Match Accuracy	POS Match Accuracy	First Word Match
1	0.02	3.6	1.8
2	0.08	1.9	1.6
3	0	3.1	0.7
4	0.13	6.0	3.3

Agreement

Model #	Word Match Accuracy	POS Match Accuracy	First Word Match
1	0.02	17.5	1.8
2	0	14.0	0.01
3	1.45	10.4	11.9
4	0	11.4	0.01

Analysis

We find that the models by and large learn the incorrect rule for hierarchical generalisation. They seem to learn some variant of the linear rule instead. One interesting observation is that in general, difference between POS match and word match on test data is lower for the agreement language than the no-agreement language, which might indicate how agreement improves learning. This has to be subjected to more rigorous statistical testing to confirm.

Some examples of the kinds of errors the models make are:

Expected: Doesn’t her monkey who does live call the elephants?

Predicted: Does her monkey who doesn’t call the elephants?

Expected: Does our elephant who doesn’t giggle impress our dogs?

Predicted: Doesn’t our elephant who does giggle does impress our dogs?

Expected: Will the seal who can live impress her seals below her dogs?

Predicted: Can the seal who will live will impress her seals?

Expected: Would your dogs that the yaks could read irritate the dogs?

Predicted: Could your dogs that the dogs could would irritate the dogs?

These results are not the same as the findings of McCoy et al, who find that the GRU with attention architecture performs better than other architectures and actually manages to generalise correctly.

This might be due to different reasons. One of them might be that we haven't trained enough models to find the initialisations that work. McCoy et al note that the initialisation of the model matters, and that results can vary significantly by intialisation. Note that model 3 for the agreement language has much higher accuracies than the others, which may indicate that that was a more suitable initialisation.
It could also be due to overfitting, which we can try to remedy by increasing dropout in future attempts to train the model. The difference in results could also be due to the way the training data was generated, and some patterns in the data we generated that we could not spot.

Conclusion

Our model did not perform the same way as the model of McCoy et al. These could be because of various reasons :

- We could've been very unlucky with our intitalisations and consistently hit up on bad initialisations. Our results varied wildly depending on the intialisations and hence with more models we might be able to get the same reports as McCoy et al
- Our data set was slighlty different, and our grammar might have been different and all this might have contributed to the different results.

- There might have been architectural optimizations which we were not privy to
- As our model consitently learnt the linear rule, and even to McCoy et al only the GRU with attention model performed better. Thus there could be a heirarchial bias.

Future Work

- Use a linear classifier to test the final encoder states, to get a better understanding of what the network is actually learning
- Compare the errors found with the findings of Crane and Nakayama
- Try this on more complex hierarchial tasks
- Try this on real world data, instead of generated data
- Try this with other languages