Chapter 1

Conclusion

Many aspects of the OceanQA dataset and the H-PERL approach have been discussed in previous chapters. This chapter therefore aims to summarise these details, and particularly focus on the strengths and weaknesses of our VideoQA architecture. We also attempt to compare our approach qualitatively to existing work in VideoQA. Finally, we conclude by outlining a number of possible extensions to our architecture, some of which may help to address the drawbacks of the approach, or apply the approach to different tasks.

1.1 Summary of H-PERL

In Chapter **??** we described the H-PERL architecture as a hybrid architecture - composed of both neural networks and symbolic reasoning modules - made of a pipeline of components. Each component in the architecture extracts a particular concept from the video or question. These concepts include: objects, including type and position; object properties, such as colour and rotation; binary relations between objects, such as close; and events between two consecutive video frames.

Chapters **??** and **??** each outlined specific H-PERL models. The hardcoded model, outlined in Chapter **??**, used mostly manually engineered components. These components allowed the hardcoded model to find and correct errors in the object detection, but also made it less adaptable to new environments. The trained model, described in Chapter **??**, continued to use some engineered components, but trained its *core* components using QA pairs. The trained model also proved the usefulness of hybrid models, by allowing background knowledge of the environment to be used to find the set of effects which occur in a video.

In Chapter **??**, which outlined the performance differences between the models, we saw that the trained model was significantly faster than the hardcoded model, and that the error correcting abilities of the hardcoded model made it slightly more accurate. We also noticed that both models we heavily reliant on highly accurate object detection, but that the trained model was less susceptible to a reduction in accuracy due to noisy data.

Chapter **??** noted that one of H-PERL's key strengths was its adaptability to other tasks; as long as there exists QA pairs with which a component can learn a particular concept, many different QA tasks can be solved. Chapter **??** also mentioned a number of requirements that each H-PERL model has, as well as a number of assumptions that needed to be made about the data. These requirements and assumptions could be deemed quite strict, however, future extensions to our architecture may be able to overcome them.

We now summarise the weaknesses of the current state of the H-PERL approach that have been encountered throughout this report:

1. Both H-PERL models require a highly accurate object detector, which, for many environments, is a very difficult requirement to meet.
2. The H-PERL architecture relies on a hardcoded question parser, which limits each model's ability to adapt to new environments, since, in the general case, parsing questions by-hand is almost impossible.
3. Currently, H-PERL relies on datasets containing specific questions which can be used to train components individually. In a general question-answering setting this will not be the case; questions will require knowledge of many different concepts to answer.
4. H-PERL can currently only solve environments which require knowledge of only: objects, object properties, relations between objects and events. This assumption is too simplistic for many environments.
5. Unlike other machine learning methods, an environment specification must be provided to each H-PERL model. For complex environments, this may be very difficult to generate.

Of course, H-PERL comes with many advantages as well. We now list the following strengths of the H-PERL approach:

1. Firstly, H-PERL is highly accurate on the OceanQA dataset; with one of the models answering 99.1% of questions correctly.
2. Since H-PERL stores information extracted from a video symbolically, it is very simple to analyse, understand and explain the model's behaviour.
3. The hardcoded H-PERL model was able to correct errors in an earlier component by reasoning about uncertainty. And, indeed, there are many more possibilities for further symbolic error correction functionality to be added, since H-PERL represents information about each video explicitly.
4. Both H-PERL models proved the utility of background knowledge injection. The hardcoded model was able to use an entire $\mathcal{AL}$ model to detect events, and the trained model was able to use ASP rules to detect effects.

5. After symbolic information has been extracted from the video, all of the questions for that video can be answered in one go, meaning the architecture can be quite fast. This is in contrast to some VideoQA implementations which make use of attention; each question is applied to the attention module individually in order to extract the relevant parts of the video.

1.2 Comparison with Existing Models

1.3 Future Work