The aim of our project was divided into 2 major aspects:

1. To be able to identify the relevant multi-hop answer sentences in the Multi-RC Dataset
2. To do a comparison of some of the available pre-trained sentence embeddings

In order to complete the first objective, we had also formulated some theory of using Graph Neural Networks, in combination with pre-trained embeddings like the Universal Sentence Embeddings, and implementing the process in [citation] (ComQA), which would speed up the training, as compared to generating embeddings from scratch with BERT. However due to the novelty of this field, and lesser available resources on GNNs, we had not been able to implement the process, and took another approach, to see if sentence embeddings themselves could identify the correct answer sentences.

So in our modified approach, we tried to see if sentence embeddings could be used to identify the correct answer sentences for the questions for each comprehension in the Multi-RC Dataset.

To accomplish this, we formulated 2 processes:

1. Encode all the answer sentences and the question sentences, and use a dot product or cosine similarity, with normalisation, to identify which answer sentences are the most similar to the question sentences, and based on number of answer sentences (Assume ‘K’), see if the top ‘K’ most similar sentences are in the answer sentences
2. We also tried to enrich the embeddings with Gated Convolutions, as tried in [citation] (Website), where we generated Knowledge Graphs, and for each comprehension, the embeddings are aggregated with their neighbours, explained more in detail in the implementation.

We also made some metrics to help evaluate how well the embeddings were able to identify the answer sentences, and if a process like this is feasible enough to predict answers for multi-hop questions.

Here are the Pre-Trained Sentence Embeddings we used, to conduct our experiments:

1. Universal Sentence Encoder (USE) Embeddings from Google
2. InferSent Embeddings from Facebook
3. Siamese BERT Embeddings

For generating the Knowledge Graphs, we relied on the process done in [citation] (Website), and modified their process, to improve connections in the graph. The detailed process can be viewed in the code available in our Github repository. Here are the steps in brief:

1. Use Google Entity Extractor with Google Natural Language API
2. The entities are extracted from a sentence
3. Lemmatization & Stop Words are removed as Nodes
4. If the entity is a Wikipedia Entry, pull WikiData
5. In Case of No Entities from Google, use Nouns & Proper Nouns
6. To create the Knowledge Graph for each comprehension:
   1. Each sentence in the comprehension is seen as an article, which invokes the entity extractor
   2. For the list of entities we get, it creates an entity node for each one
   3. For those entities from Wikipedia, it creates a list of “Instance Of” nodes
   4. Then it creates an article node, denoting the sentence here
   5. At the end, all edges are added

We will describe the metrics in detail in the implementation, but here is the intuition behind the metrics we used to evaluate the results:

1. An accuracy like metric, which takes into account the number of question and answers in each individual comprehension
2. Above accuracy, with just missing 1 sentence in the answer sentences
3. Accuracy with 1 prediction relaxation, where we see if the ‘K’ answers are there in the top ‘K+1’ similar sentences, instead of top ‘K’

The experiments conducted by us were not to surpass any bench marks on datasets, but to see if pre-trained embeddings can be used to predict multi-hop sentences in question answering, and our work could be extended in future to incorporate GNNs with pre-trained embeddings for Multi-Hop Question Answering.

\begin{tabularx}{0.8\textwidth}{

| >{\centering\arraybackslash}X

| >{\centering\arraybackslash}X

| >{\centering\arraybackslash}X

| >{\centering\arraybackslash}X

| >{\centering\arraybackslash}X | }

\hline

Embeddings & Custom Accuracy & \%Questions Correct or 1 Missed & Accuracy with 1 Prediction Relaxation & \%Questions correct or 1 missed with relaxation \\

\hline

USE & 52.2\% & 74\% Qs & 62.1\% & 81\% Qs \\

\hline

InferSent with updated vocabulary & 48.76\% & 70\% Qs & 57.84\% & 78\% Qs \\

\hline

InferSent with standard vocabulary & 48.65\% & 70\% Qs & 57.2\% & 77\% Qs \\

\hline

Siamese BERT & 45.2\% & 66\% Qs & 54.6\% & 74.5\% Qs \\

\hline

\end{tabularx}