**General Information:**

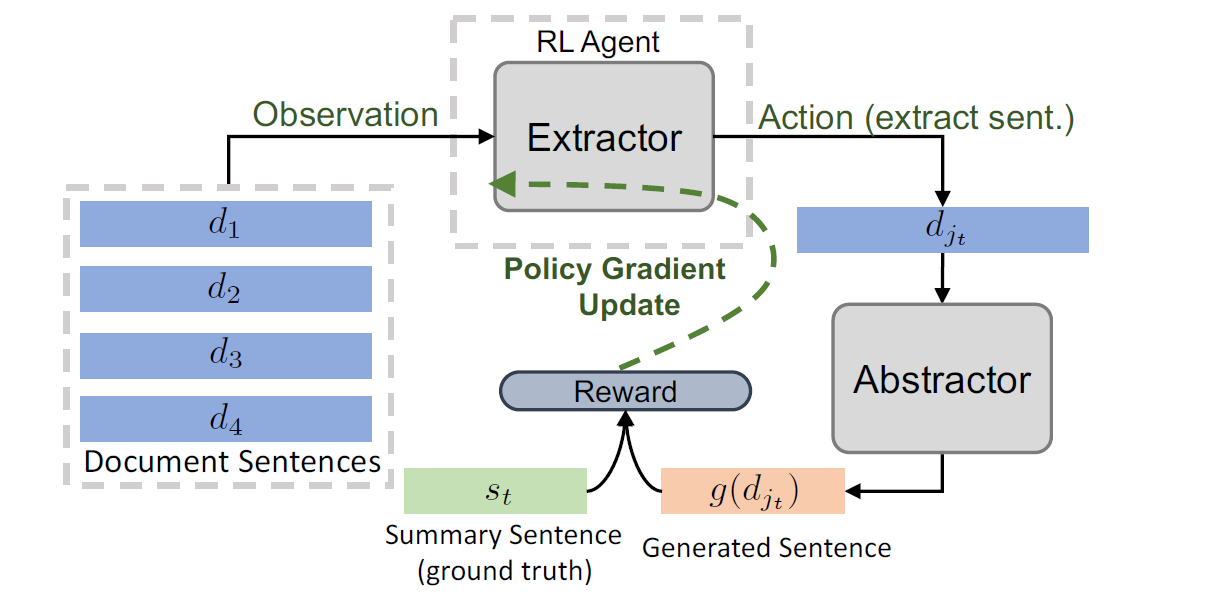
**This model is based on the ACL2018 paper** [*Fast Abstractive Summarization with Reinforce-Selected Sentence Rewriting*](https://arxiv.org/abs/1805.11080).

The algorithm architecture came from <https://github.com/chenrocks/fast_abs_rl>, but I modified it so that it can directly give the summarization of articles.

The model is trained on a set of CNN and Daily Mail stories (which contains 287k news articles with corresponding summarizations).

How to run this model: please read the readme.md in the model’s repo.

**Architecture details:**



This supervised learning model trains the Extractor (to extract important sentences from a news article) and Abstractor (to rephrase the extracted sentences) separately, and use reinforcement learning (Policy Gradient) to optimize the performance of the whole model, so that it can generate a good summary of the news article.

It also re-rank the sampling of sentences using a beam size such that there’s no replicated sentences in a summary.

For further details, please consult the paper [*Fast Abstractive Summarization with Reinforce-Selected Sentence Rewriting*](https://arxiv.org/abs/1805.11080).

**Pros and Cons:**

Pros:

* It is the state of the art text summarization model.
* It is a universally worked model, which means it should be able to summarize any document with the pre-trained weights.
* The inference is fast. It only takes several milliseconds to summarize an article with hundreds of words (if the model is running on azure NC6 virtual machine)

Cons:

* It is a supervised learning algorithm, which means that if we want to achieve the best performance on a specific type of data (for example, safety reports), we need to provide the model sufficient document-summary pairs so that it can learn the architecture of safety reports and make sense of meanings of words and sentences in safety reports (e.g. the model doesn’t know the meaning of “oil damper” until it reads about it, just like how human being make sense of articles.)

**Experiments:**