Project Report – Planetary IR

Aim:

To develop an application using DeepDive to automate extraction of authors, targets and compositions from planetary science papers to help understand and develop an encyclopedia for Mars.

Challenges:

The main challenges faced while building this application are stated below:

* Planetary Science papers are formatted in a different way with title and authors having a superscript of numbers which indicate the organization that they belong to
* NER systems are not trained against these standards and hence failed to recognize authors as PERSON objects
* DeepDive is not built to currently support incorporating trained models which vary from the standard English models that they already use
* Martian targets do not have a naming convention and can range from person names to location names. There is no heuristic which will tell us what targets are i.e. there is currently no exact definition for them.

Current Solution:

Note: DeepDive does not use machine learning to feedback useful features and modify the extractions by itself. Instead, it uses an inference and learning engine to indicate how well the extractions are doing and which features are particularly useful. This way the developer can perform error analysis and debugging to improve the application iteratively by looking at the statistical inference results and the various calibration plots.

The current solution is built as a DeepDive application to extract authors and targets. The structure of a DeepDive application is explained here: <http://deepdive.stanford.edu/deepdiveapp>. The root of my application is app/planetaryir. In addition to this, the repository contains a folder called scripts which hare required for pre-processing and post-processing. Below is a description of those files:

1. textConvert.py – Script to convert the text files into a json list file which is DeepDive’s input format. Whenever we have new text files, we can use this script to convert all the text files in the Samples folder into a json list file. The json list will then be taken and copied over to the input/ folder of the application.
2. PdfParse.java – Program to convert PDF to text using Apache Tika. This is required if we only have sample documents in PDF format.
3. check\_target.py – Used to perform evaluation of targets, this script breaks the text files into single tokens and compared each word to the known target list and produces an output of all the potential targets that were found in that document.