# Progress Report

1. **Which tasks have been completed?**

   **Filtering: Python script**
   - ➔ Read DBLP.xml dataset and only select relevant articles for the experiment, e.g., 'Data Mining' articles.
   - ➔ Remove punctuation marks in the titles except when they are part of the word, e.g., "Mr." or "can't".
   - ➔ Stem words using the Krovetz stemmer as suggested in the original paper.
   - ➔ Print the pre-processed articles to an XML file and also print the titles and authors in a format that is readable by the SPMF tool.

   **Closed Frequent Patterns: SPMF tool**
   - ➔ Run SPMF tool to generate closed frequent patterns for the authors, using FPClose, and the titles, using CloSpan, as suggested in the original paper.
   - ➔ Run a UNIX command to clean-up the output files produced by the SPMF tool so that the next stage in the pipeline can read them.

   **Cluster Frequent Patterns and remove redundancy: Python script**
   - ➔ Read Closed Frequent Patterns and run the One-pass Micro-Clustering algorithm described in the paper.
   - ➔ Print the medoid pattern of each micro-cluster and report the compression rate, i.e., how many patterns were removed.

   **Semantic Annotation: Python script**
   - ➔ Read the list of compressed Frequent Patterns from the previous stage and given a pattern query, calculate the Mutual Information of each pattern with the query and select a list of context indicators.
   - ➔ Given the list of context indicators, select the Frequent Patterns that are semantically similar to the query.
   - ➔ Given the list of context indicators, select a list of representative transactions that share a similar context to that from the query.
   - ➔ Print the query pattern together with its context indicators, a list of semantically similar patterns and a list of representative transactions.

2. **Which tasks are pending?**

   All the steps described in the paper have been implemented and the only remaining tasks right now are:
   - ➔ Integrate all the stages into a single runnable script for user convenience. Right now you have to manually run each stage of the pipeline.
   - ➔ Improve the quality of the semantic annotations. Currently, some of the semantic annotations I output are meaningless patterns or seemingly unrelated transactions, I

don't know if this is because of the quality of the patterns I'm selecting or some bug in my code.

➔ Finish the documentation and prepare the demo presentation once I'm able to generate good results.

**3. Are you facing any challenges?**

Like I mentioned in the previous question, I'm not able to generate good enough semantic annotations for my input queries, this might be due to:

- ➢ Selecting articles from the DBLP dataset that do not represent a topic well: Currently, I'm filtering articles by journal which might not be the best discriminative field.
- ➢ Input data needs more clean-up: Many of the patterns I'm observing for the titles consist of stop words, e.g., "of a", which add a lot of noise to future stages in the pipeline. The paper doesn't explicitly say I should remove these words, on the contrary, it argues that the algorithms it describes are robust against this specific type of noise so maybe something else is wrong.
- ➢ The stage that selects Closed Frequent Patterns is faulty: It's unlikely that the third party tool I'm using to run FPClose and CloSpan is faulty (it's from a popular toolset that is heavily tested) but there might be a bug in my code that clusters patterns and removes redundancy. Although, overall, I think it's unlikely that bad pattern selection alone is enough justification for poor semantic annotations, this is because not all the patterns I'm observing are bad so the rest of the pipeline should at least work for them.
- ➢ The stage that annotates patterns is faulty: Perhaps there is a bug in my code for this particular stage.

I will have to spend more time probing and debugging each step of the pipeline with small test cases to validate that my implementation is correct.