

<CS410 : Project Progress>

Project Topic

- Project topic : (Option 1) Reproducing a paper: Latent Aspect Rating Analysis

Project Schedule (Table 1)

Date	TODO
11.20 ~ 11.28	Paper review
11.29	Progress report
11.30 ~ 12.01	Source review
12.02 ~ 12.08	Source implementation
12.09 ~ 12.11	Result analysis & Documentation
12.12	Source code & Documentation submission

Team members

1. Hongseok ha (netID : hh23), with administrative duties
2. Changsoo Kim (netID : ck37)

Progress Report

1. Which tasks have been completed? (Progress made thus far)
 - A. Paper review
 - i. We went through the paper to get a holistic understanding and found that this paper suggests a unified framework LARAM, which improves LARA and enables aspect rating analysis without knowledge of the target domain.
 - ii. LARAM has two components: 1) an aspect modeling module, and 2) a rating analysis module similar to the Latent Rating Regression Model (LRR) used in LARA.
 - iii. Once we infer the latent aspect assignment z and aspect weight α with the given model $\Theta = (\epsilon, \gamma, \beta, \mu, \Sigma, \delta^2)$, we can estimate corpus-level parameters using the Expectation Maximization algorithm.
 - B. Find materials that help understand the paper
 - i. We found the author's original source from his homepage. This would be very helpful to understand the process more precisely. However, it is implemented in Java, so we need to convert it to Python.

- ii. URL: <http://sifaka.cs.uiuc.edu/~wang296/>
- 2. Which tasks are pending? (Remaining tasks)
 - A. Python code implementation
 - i. As there are two components and we have two members, each member will be in charge of one module.
 - 1. Hongseok Ha: Aspect modeling module
 - 2. Changsoo Kim: Rating analysis module
 - B. An overall schedule is shown in Table 1.
- 3. Which tasks have been completed? (Any challenges/issues being faced)
 - A. Ambiguity of aspect modeling part: The aspect modeling module behaves similarly as LDA or sLDA, but it has different assumptions. We cannot spot the big difference so far, so we sent an email to the TA and we are going to ask on Piazza as well. If there are only subtle difference, we are going to use LDA instead and see the difference between the paper's result and ours.
 - B. Python portability: We will convert the author's original code in Java to Python. These differences can lead to slight differences in results.
 - C. Experiment results: In the paper, there are several comparisons, such as LDA vs sLDA vs LARAM, LDA+LRR vs sLDA+LRR vs LARAM, Bootstrap+LRR, LARAM, and so on. If we were to compare the number of all cases, it would take a lot of time. We will only compare the paper's suggesting result and our result.