Baselines - Combating Partisan Homogenization in News Recommendation Systems

Contents

4 General Experiment Settings

github-url : https://github.com/karthikshivaram24/Combatting-partisan-homogenization

4.1 TF-IDF

- News Articles Used : **127344**
- Features: **TF-IDF**
- Dimensionality Reduction : **SVD**
- Clustering Algorithm : **K-Means**
- Number of Clusters : **1000**
- Cluster Pair Filtering
  - Minimum Cluster Size : **450**
  - Minimum Partisan Size : **0.5**(used balanced sampling to make label distributions equal)
- Recommendation Model
  - Logistic Regression
  - SGDClassifier with log loss
- Performance Metrics
  - **Macro** : F1, precision , recall
  - **@K** : F1, precision, recall
- User Preferences:
  - **Homogeneous** : Likes articles of the same partisan score across cluster pair (Likes conservative articles in both clusters)
  - **Heterogeneous** : Likes articles of different partisan score across cluster pair (Likes conservative articles in cluster 1 and liberal articles in cluster 2)

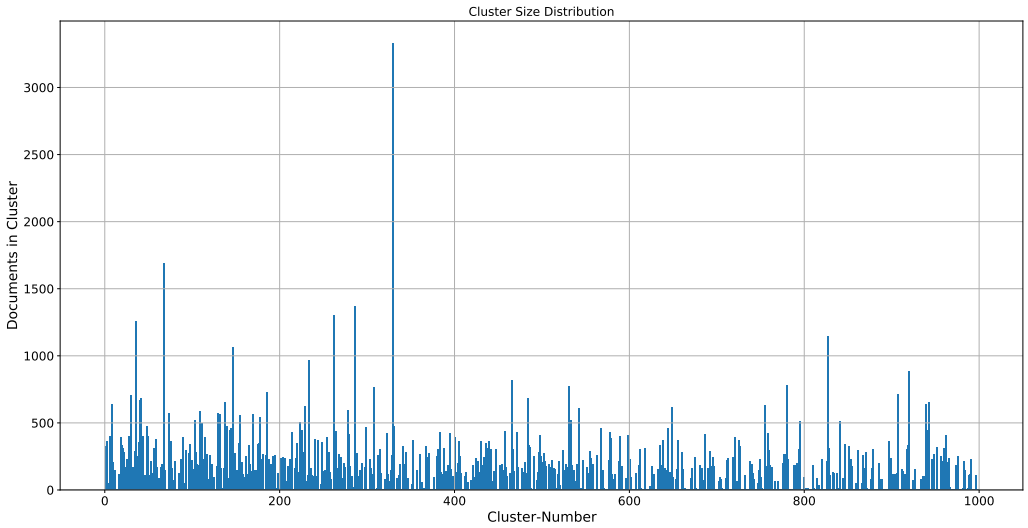


Figure 1: Cluster Size Distribution

4.2 GLOVE

4.3 BERT

5 Baseline 1: How Topic Similarity affects Model Performance

Here we want to measure how well a recommendation system performs on an unseen topic for users with different types of preferences and how this performance varies as similarity between topics increases (seen and unseen topic similarity).

5.1 TF-IDF

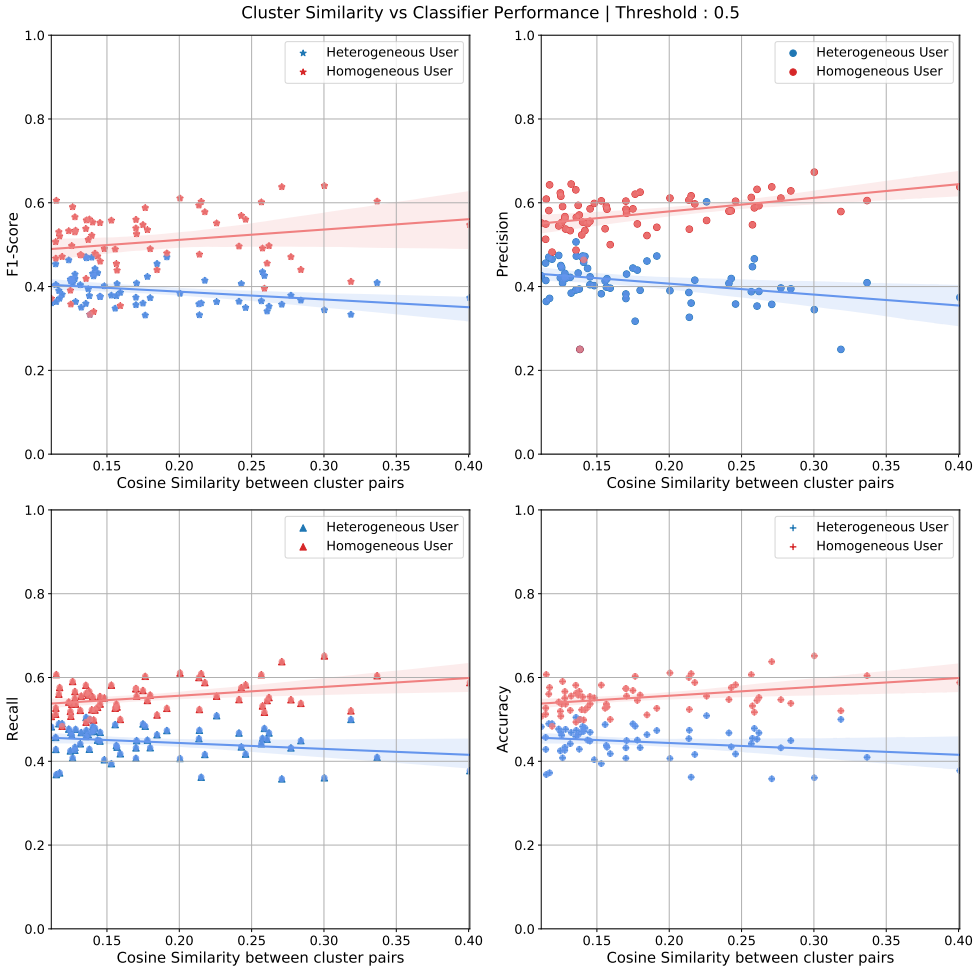


Figure 2: **Cosine Similarity between Cluster/Topic Pairs vs F1, Precision, Recall and Accuracy.The models are trained on cluster 1 and metrics are measured on cluster 2 which is used as a test set** - TFIDF

Note : The metrics measured for this baseline experiment is based on a test set and not calculated at a particular time step or user-interaction

- We see that for homogeneous user's there is an increase in **Precision** and **Recall** when similarity between the clusters/topics increases.
- For a Heterogeneous User we see the opposite effect with a decrease in both **Precision** and **Recall** as similarity between cluster pairs increases (hence harder for the classifier to distinguish between positives from cluster 1 and negatives from cluster 2 when both are more similar).

5.2 Glove

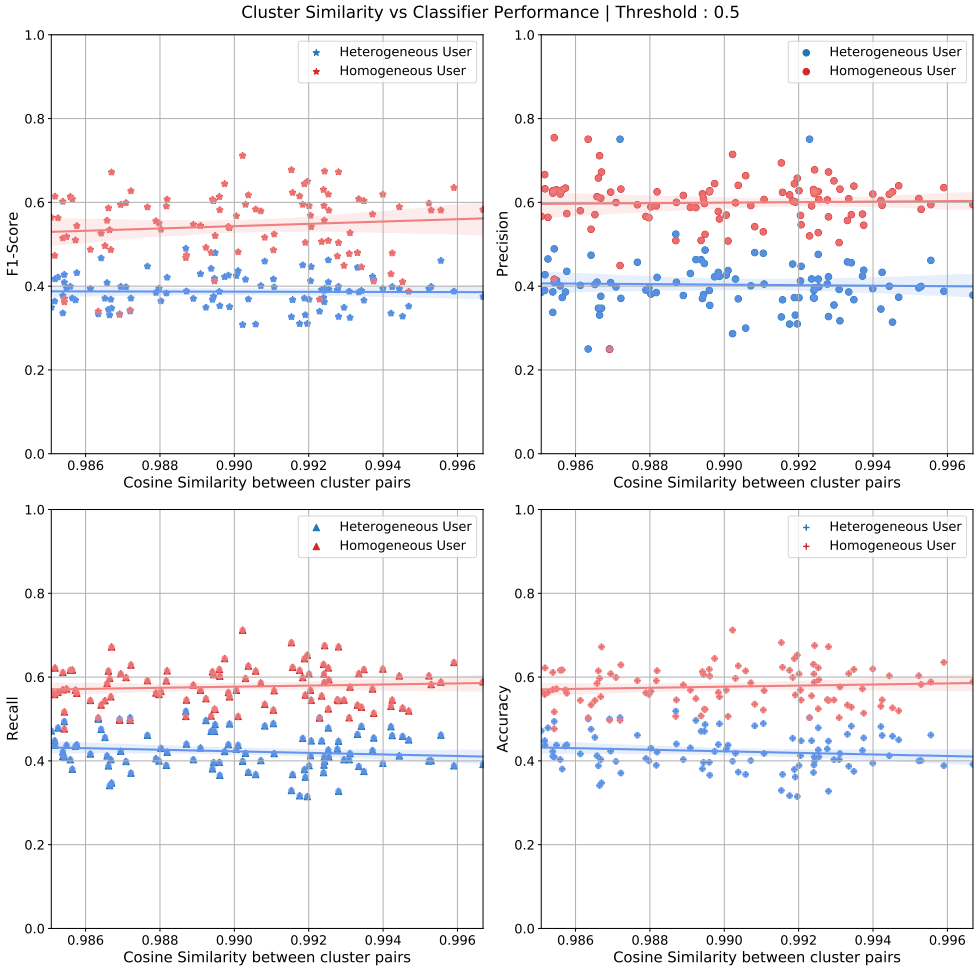


Figure 3: **Cosine Similarity between Cluster/Topic Pairs vs F1, Precision, Recall and Accuracy.The models are trained on cluster 1 and metrics are measured on cluster 2 which is used as a test set** - GLOVE

- We see that for homogeneous user's there is an increase in **Precision** and **Recall** when similarity between the clusters/topics increases (only slight increase occurs here as the cluster pairs seem be more similar here compared to clustering using TFIDF).
- For a Heterogeneous User we see the opposite effect with a decrease in both **Precision** and **Recall** as similarity between cluster pairs increases (but smaller decrease compared to using TFIDF).

5.3 BERT

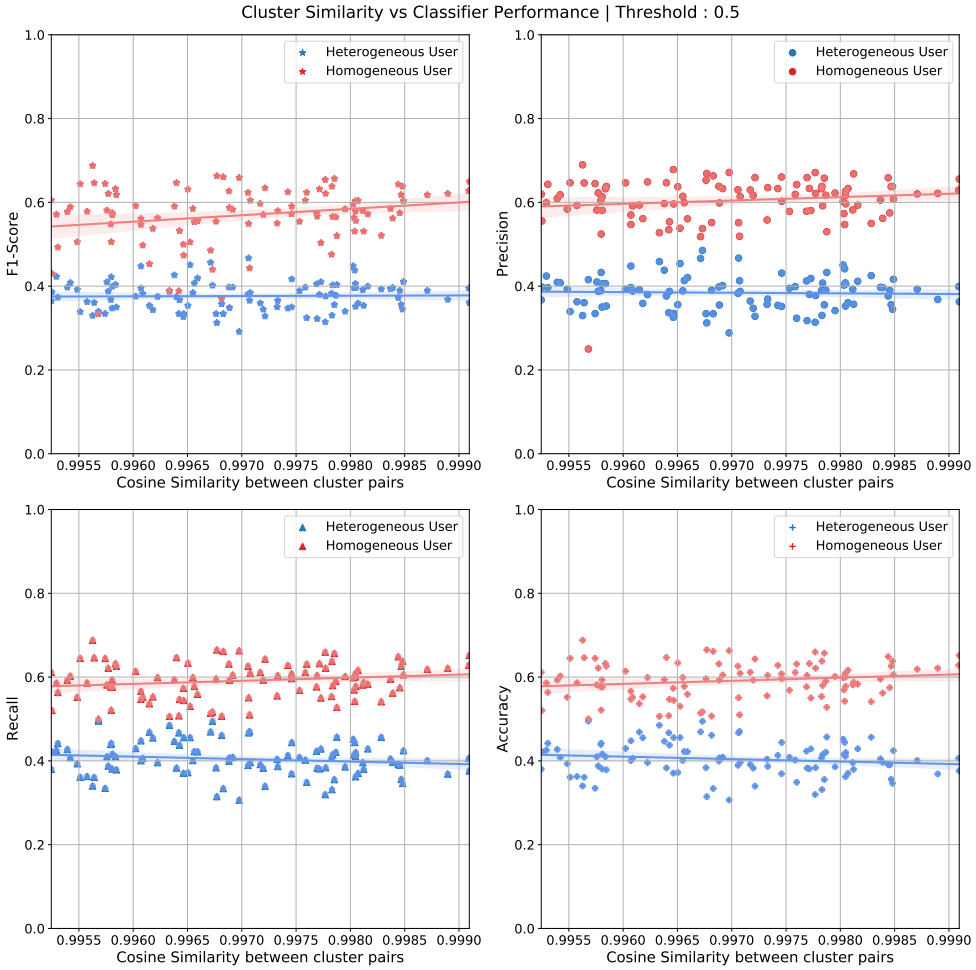


Figure 4: **Cosine Similarity between Cluster/Topic Pairs vs F1, Precision, Recall and Accuracy.The models are trained on cluster 1 and metrics are measured on cluster 2 which is used as a test set** - BERT

- Similar Performance like GLOVE but the clustering of pairs is even better now as we choose extremely similar cluster pairs measured according to cosine similarity.

6 Baseline 2: Online Learning Setting

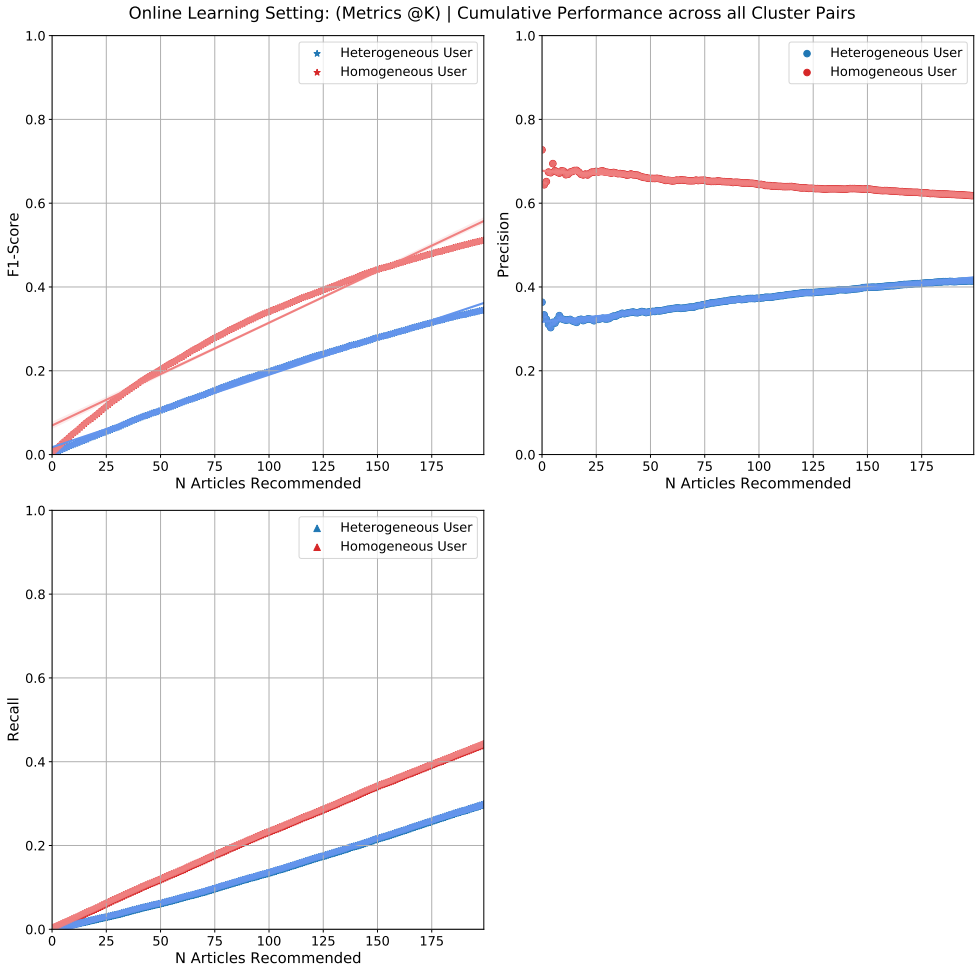
To emulate a real-world scenario , we want to simulate a recommendation system interacting with a user over a set of unseen articles , slowly updating itself to learn the user's preferences over time, **N=200** is used here to calculate the metrics, where **N** represents the number of interactions between the user and the content recommender. Also to note, there are at least N relevant articles in the candidate pool.

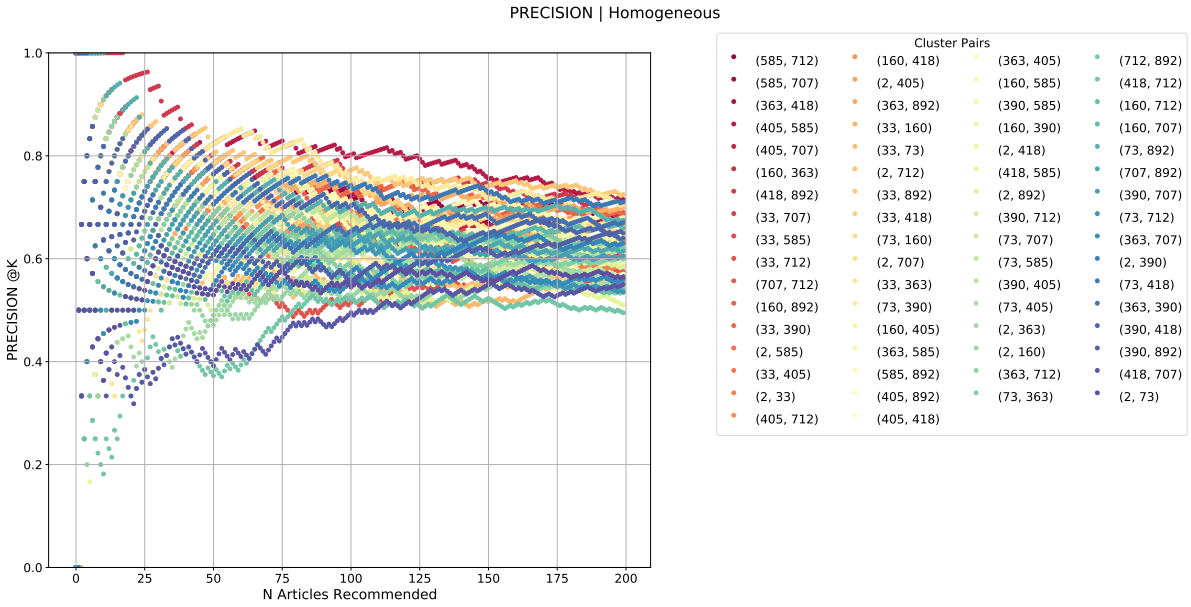
For this baseline experiment we use recommender specific metrics such as **Precision@K** and **Recall@K**.

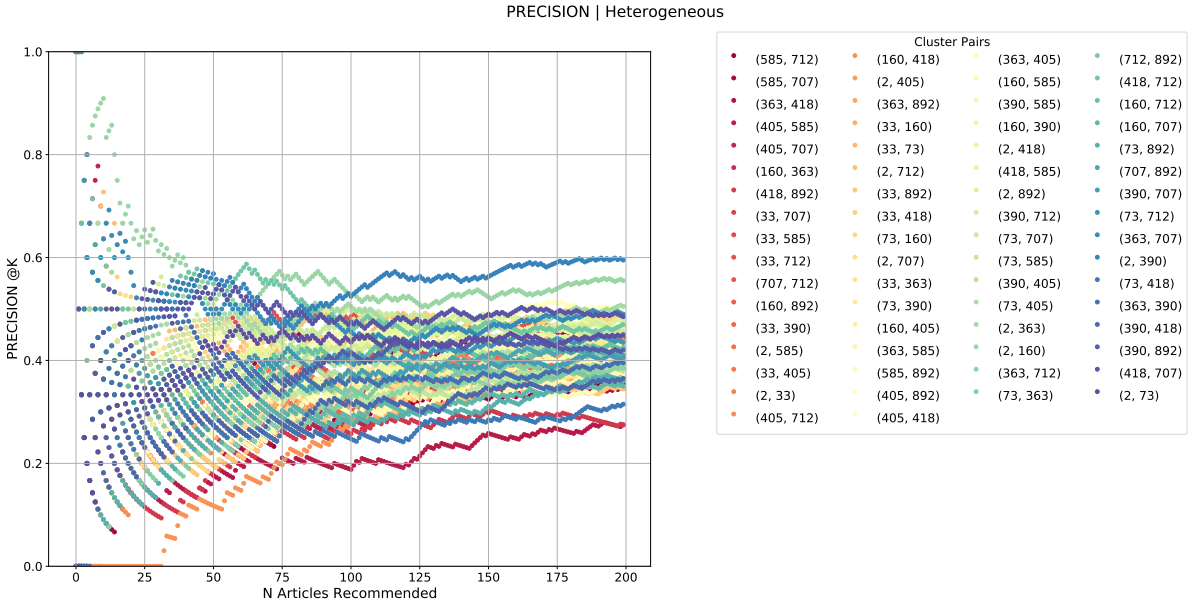
- $Precision@K = \frac{\#recommended\ items\ the\ user\ likes\ at\ K_{th}\ Interaction}{total\ \#\ of\ items\ recommended\ at\ K_{th}\ Interaction}$
- $Recall@K = \frac{\#recommended\ items\ the\ user\ likes\ at\ K_{th}\ Interaction}{total\ \#of\ relevant\ items\ in\ the\ candidate\ pool}$
- Eg:
  - Candidate Pool Size = 15
  - Number of relevant(user-likes) items in the candidate pool = 10
  - N = 1
    - Shown Articles = [0]
    - Recall = 0/10 = 0
    - Precision = 0/1 = 0
  - N = 2
    - Shown Articles = [0,1]
    - Recall = 1/10 = 0.1

    - Precision = 1/2 = 0.5
  - N = 3
    - Shown Articles = [0,1,1]
    - Recall = 2/10 = 0.2
    - Precision = 2/3 = 0.666
  - N = 4
    - Shown Articles = [0,1,1,1]
    - Recall = 3/10 = 0.3
    - Precision = 3/4 = 0.75

6.1 TF-IDF



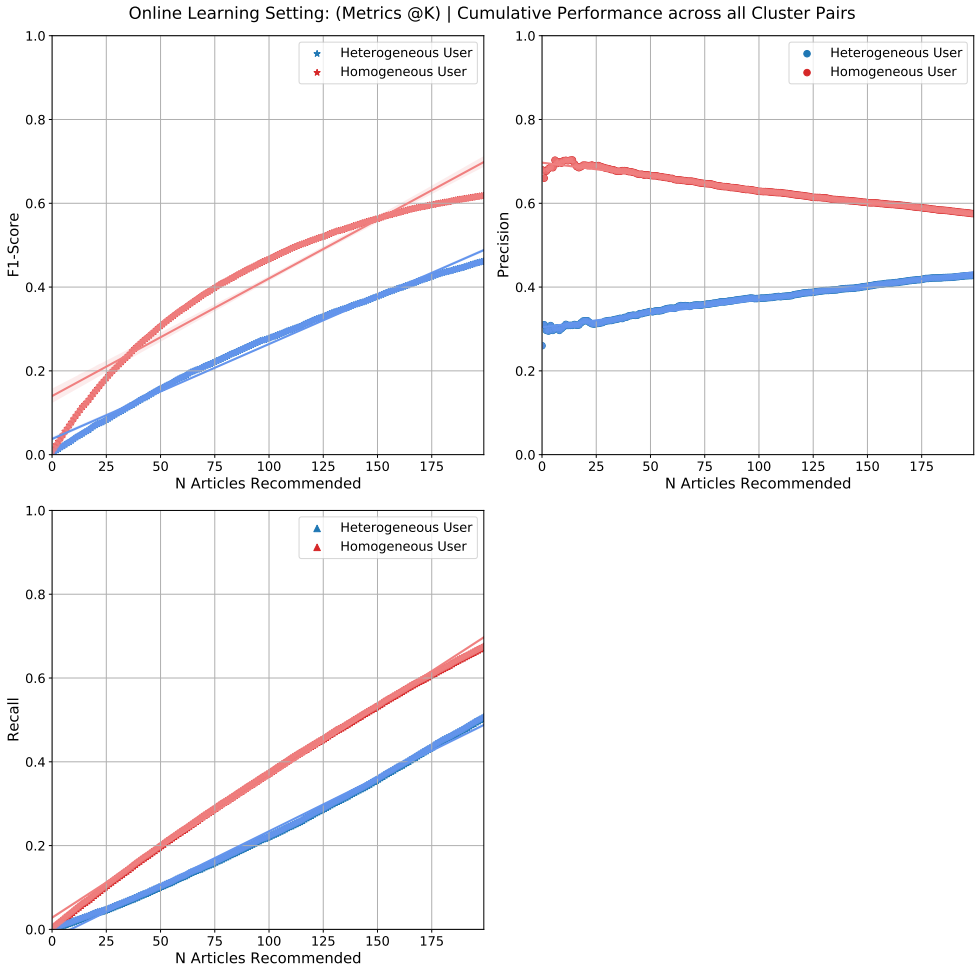


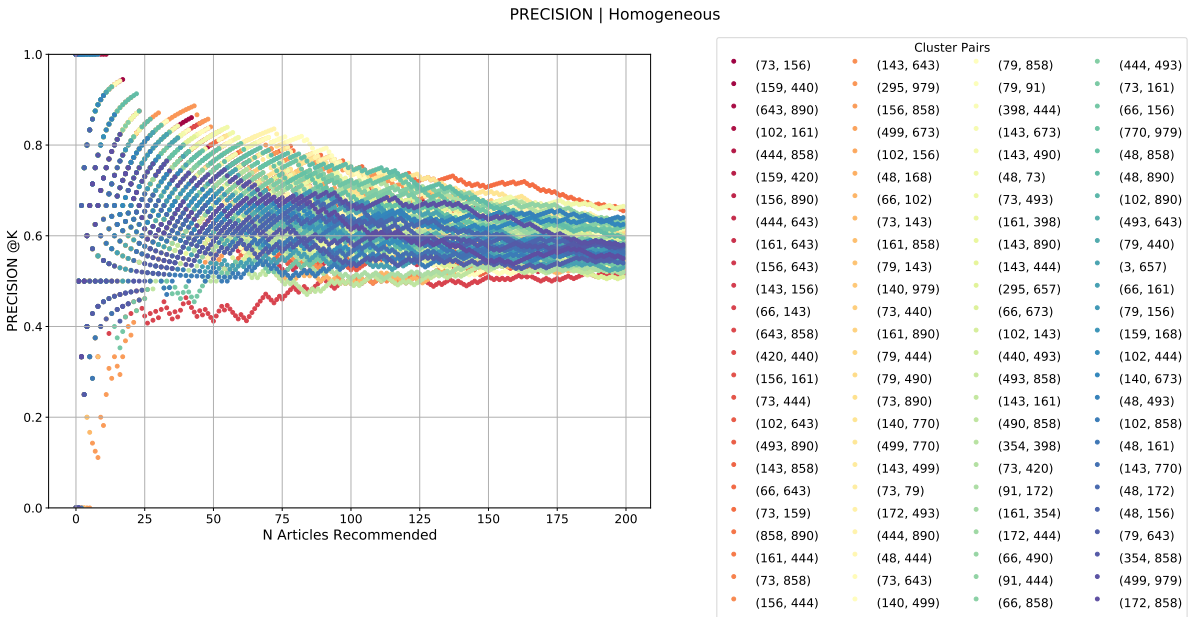


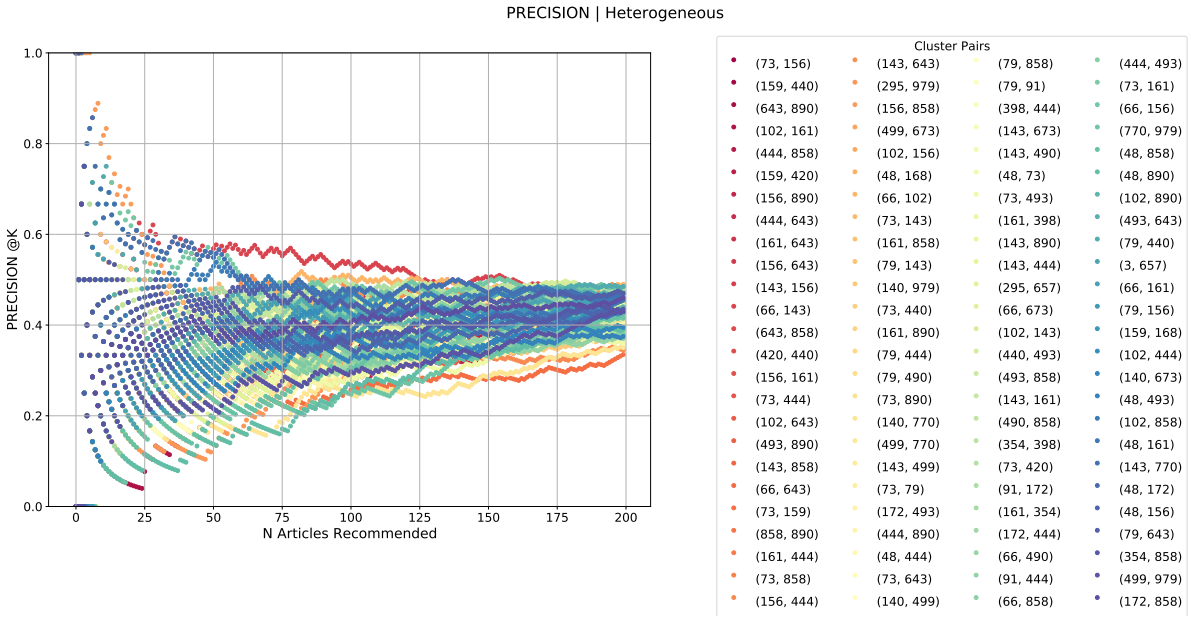
The above figure shows the mean results across all cluster pairs.

- For the Homogeneous User we see that **Precision@K** decreases as the number of interactions increase, this could be due to the fact that the most probable items the user likes are already recommended in the initial user-interactions (as we sort by predicted probability) and as interactions increase the classifier is only left with items with low confidence to recommend hence increasing mistakes. Also from the precision graph with all cluster pairs shown we see that a few cluster pairs do increase in precision.
- For the Heterogeneous User we see that their is an initial dip in **Precision@K** and then it increases as the content recommender learns the different preferences of the heterogeneous user.
- Overall we see that the precision for Homogeneous users is higher than heterogeneous users

6.2 Glove

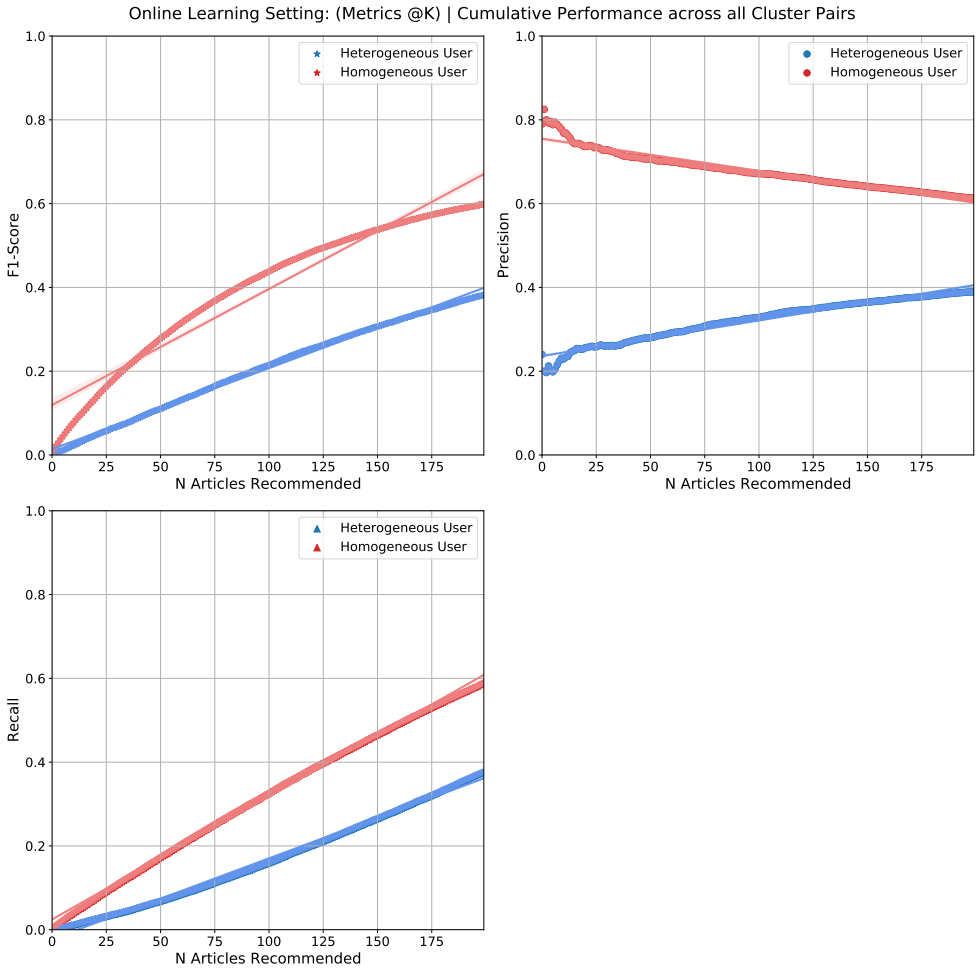

Online Learning Setting: (Metrics @K) | Cumulative Performance across all Cluster Pairs
F1-Score
Precision
Recall
N Articles Recommended
Heterogeneous User
Homogeneous User

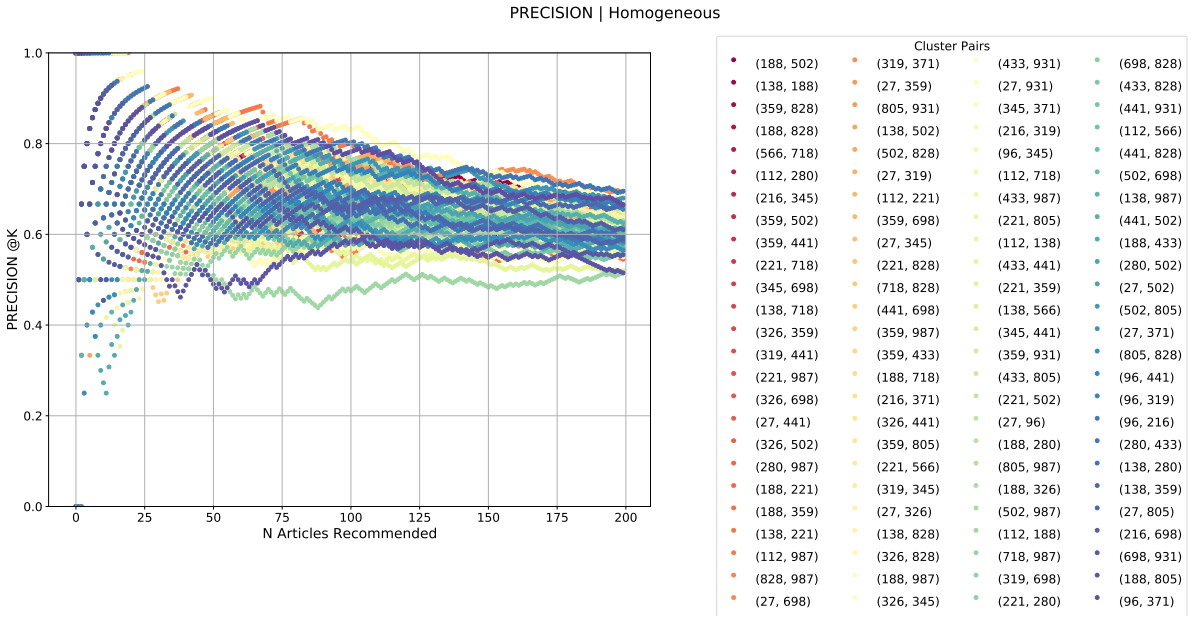

PRECISION | Homogeneous
PRECISION @K
N Articles Recommended
Cluster Pairs
(73, 156)
(159, 440)
(643, 890)
(102, 161)
(444, 858)
(159, 420)
(156, 890)
(444, 643)
(161, 643)
(156, 643)
(143, 156)
(66, 143)
(643, 858)
(420, 440)
(156, 161)
(73, 444)
(102, 643)
(493, 890)
(143, 858)
(66, 643)
(73, 159)
(858, 890)
(161, 444)
(73, 858)
(156, 444)
(143, 643)
(295, 979)
(156, 858)
(499, 673)
(102, 156)
(48, 168)
(66, 102)
(73, 143)
(161, 858)
(79, 143)
(140, 979)
(73, 440)
(161, 890)
(79, 444)
(79, 490)
(73, 890)
(140, 770)
(499, 770)
(143, 499)
(73, 79)
(172, 493)
(444, 890)
(48, 444)
(73, 643)
(140, 499)
(79, 858)
(79, 91)
(398, 444)
(143, 673)
(143, 490)
(48, 73)
(73, 493)
(161, 398)
(143, 890)
(143, 444)
(295, 657)
(66, 673)
(102, 143)
(440, 493)
(493, 858)
(143, 161)
(490, 858)
(354, 398)
(73, 420)
(91, 172)
(161, 354)
(172, 444)
(66, 490)
(91, 444)
(66, 858)
(444, 493)
(73, 161)
(66, 156)
(770, 979)
(48, 858)
(48, 890)
(102, 890)
(493, 643)
(79, 440)
(3, 657)
(66, 161)
(79, 156)
(159, 168)
(102, 444)
(140, 673)
(48, 493)
(102, 858)
(48, 161)
(143, 770)
(48, 172)
(48, 156)
(79, 643)
(354, 858)
(499, 979)
(172, 858)

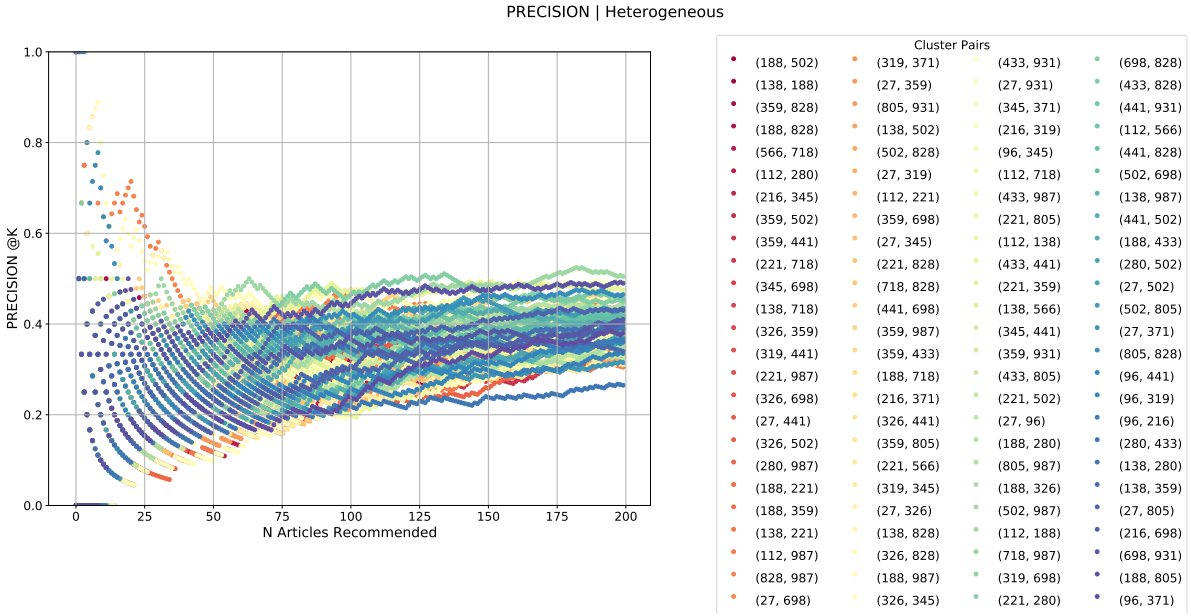


- For Homogeneous Users here we see that their is a larger decrease in Precision compared to using TFIDF representations.
- For the Heterogeneous User we see a larger increase in Precision compared to using TFIDF representations

6.3 BERT







- For Homogeneous Users here we see that their is a smaller decrease in Precision compared to using GLOVE representations.
- For the Heterogeneous User we see a smaller increase in Precision compared to using GLOVE representations

6.4 SUMMARY

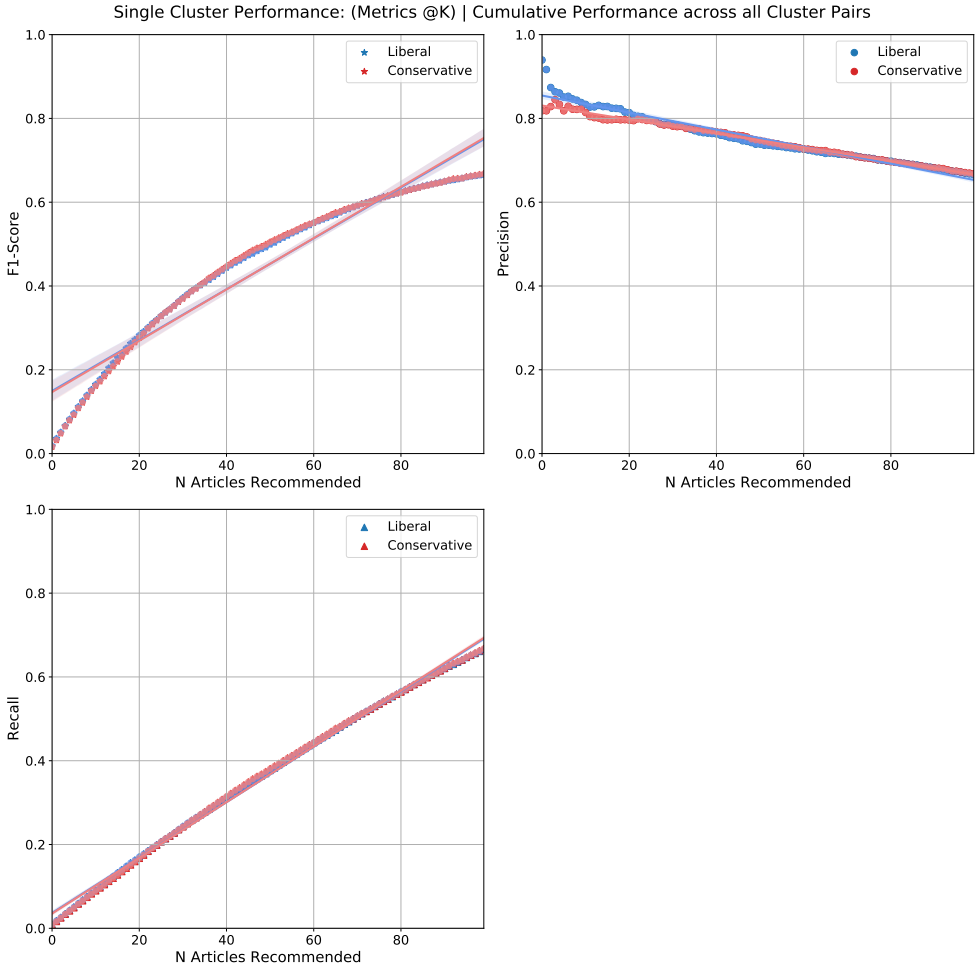
- Using static or non-contextual embeddings helps the system's performance for Heterogeneous Users (according to Precision@K)
- Using dynamic or contextual embeddings helps the system's performance for Homogeneous Users.

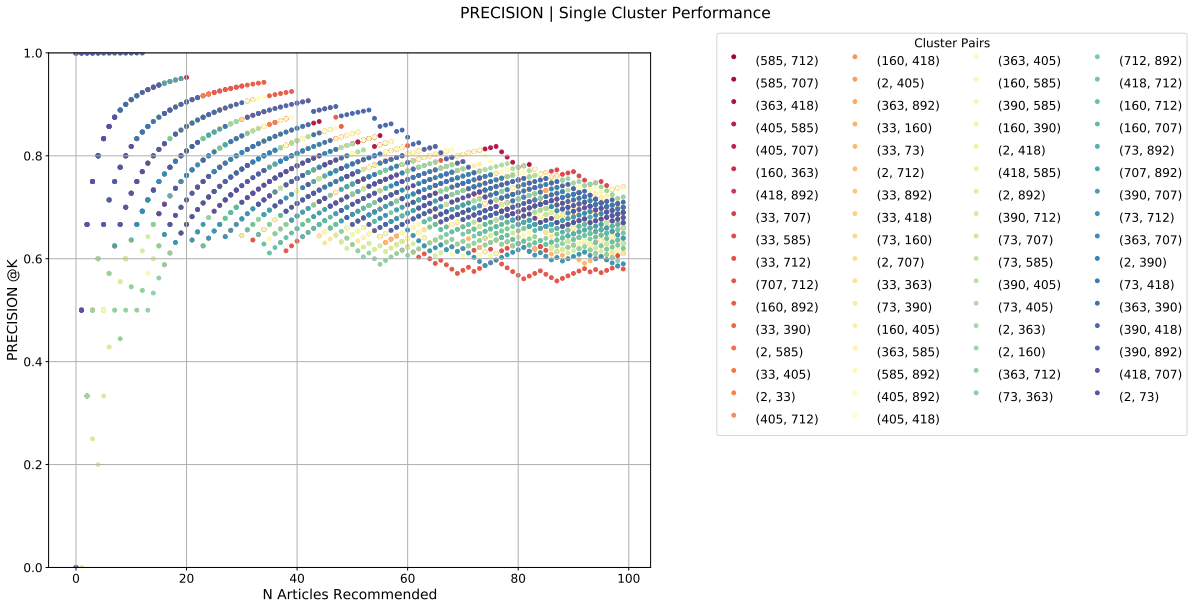
7 Baseline 3: How easy is it for the Recommendation System to detect a change in topics

We want to know how the recommendation system performs in detecting a change in topics , so we measure the performance of the system using a single cluster and compare it against our online setting performance (shown in the above baseline).

7.1 TF-IDF

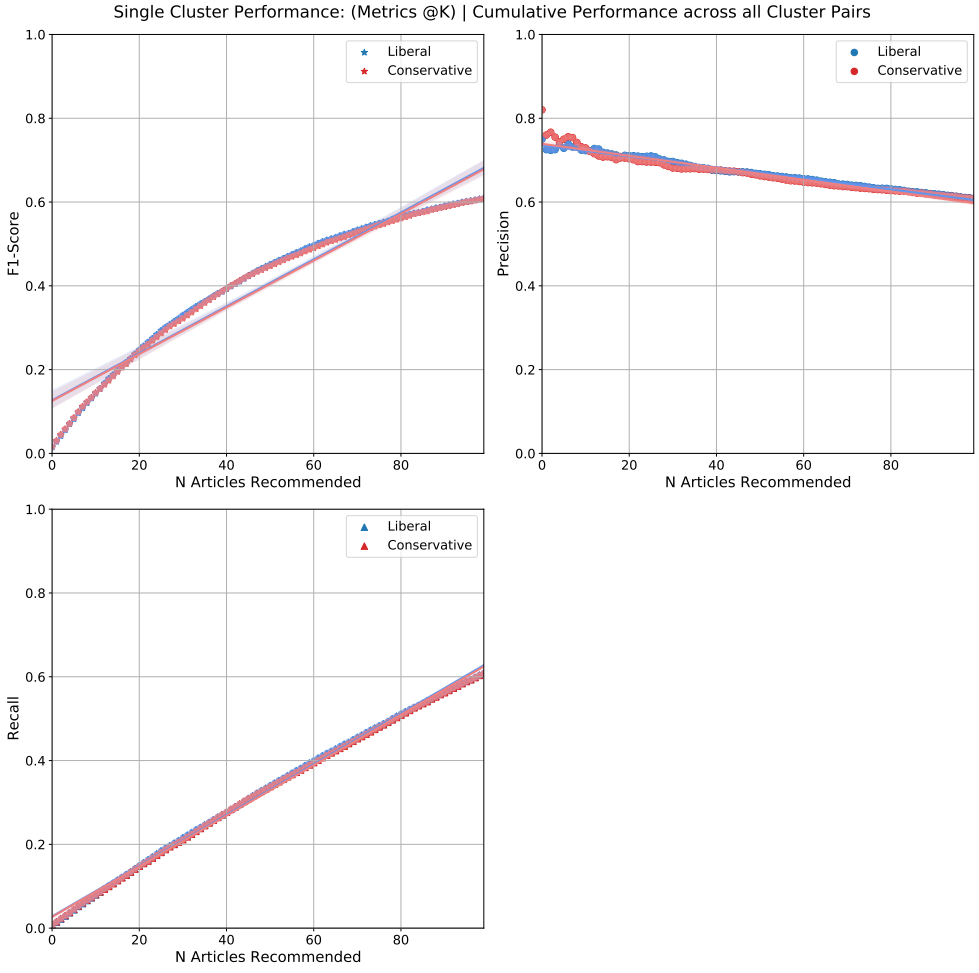
- Similar trend in Precision@K occurs here (compared to baseline 2 - Homogeneous User).
- When we compare these scores to graphs in Baseline 2 we definitely see that the recommendation system tends to have an easier time when only one topic is concerned compared to 2 different topics, so the model is having a hard time in identifying a change in topic.

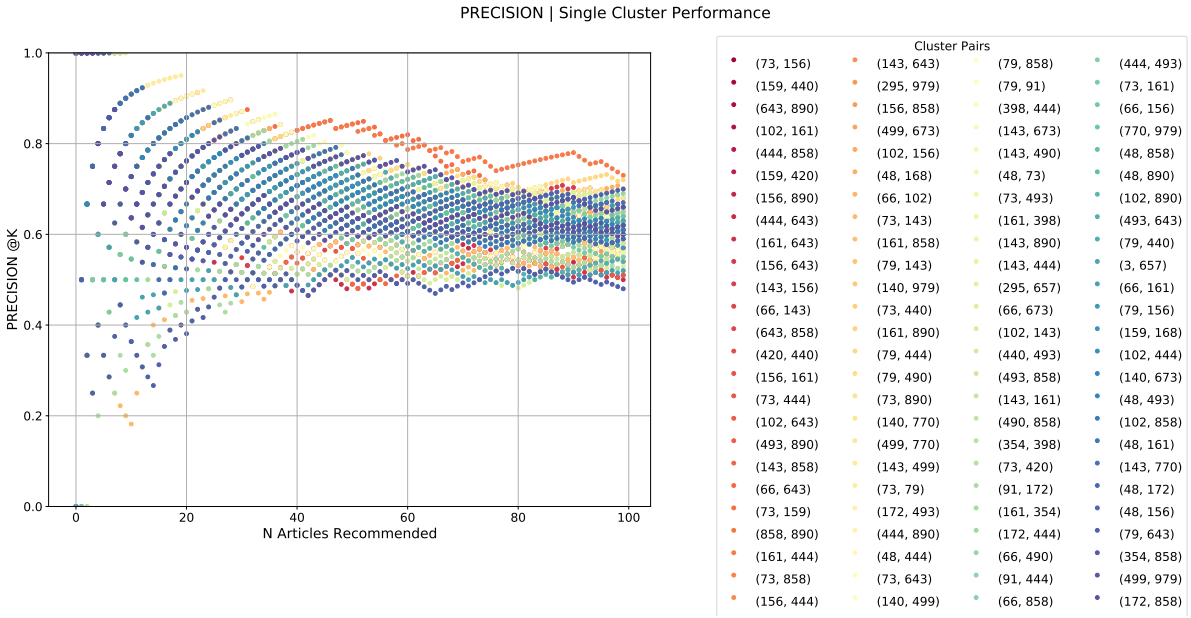




7.2 Glove

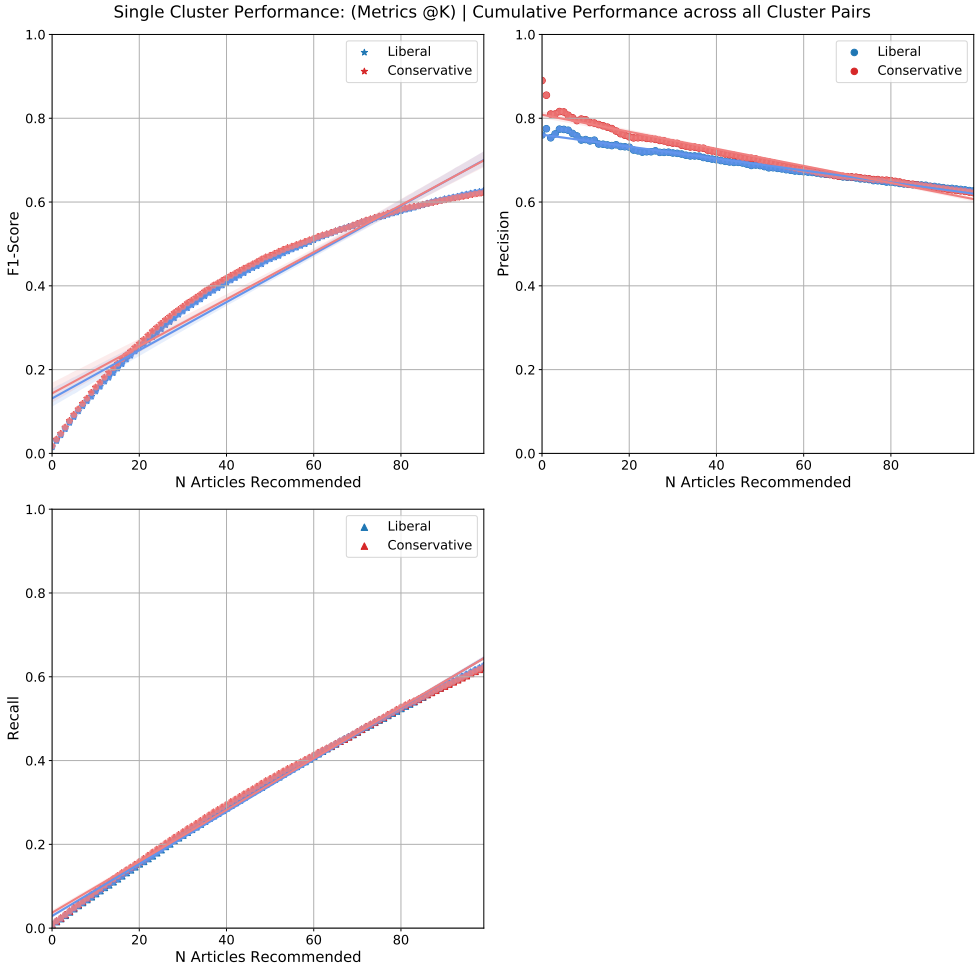
- Similar trend in Precision@K occurs here (compared to baseline 2 - Homogeneous User).
- Lower Precision compared to TFIDF Representations

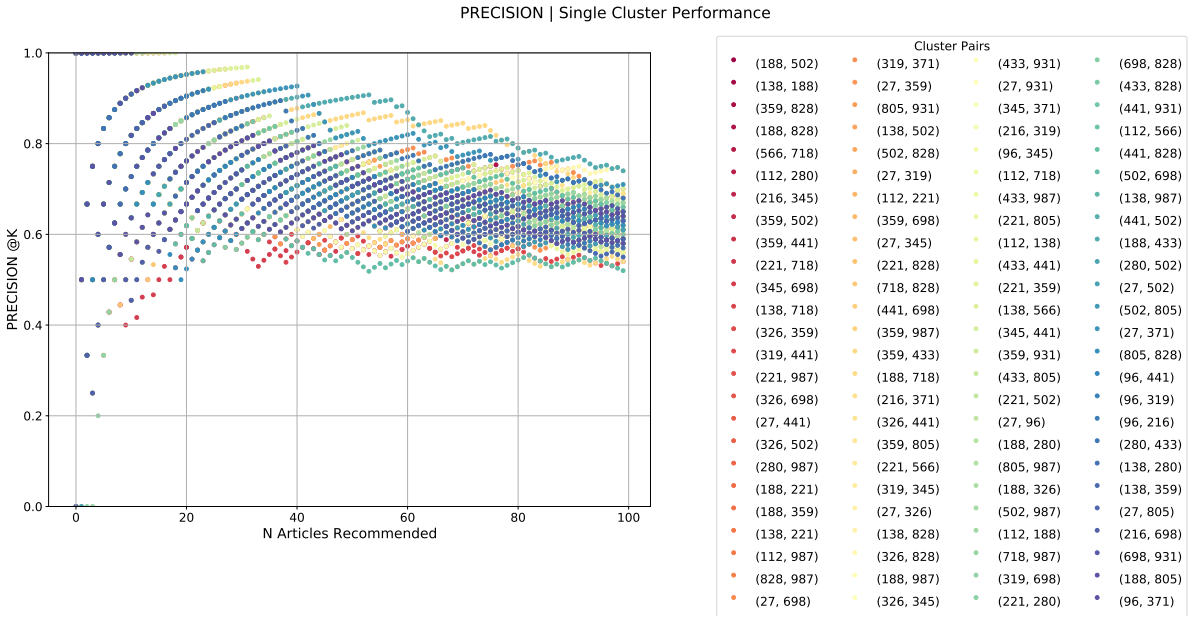




7.3 BERT

- Similar trend in Precision@K occurs here (compared to baseline 2 - Homogeneous User).
- Lower Precision trend compared to TFIDF representations but better than Glove





7.4 Summary

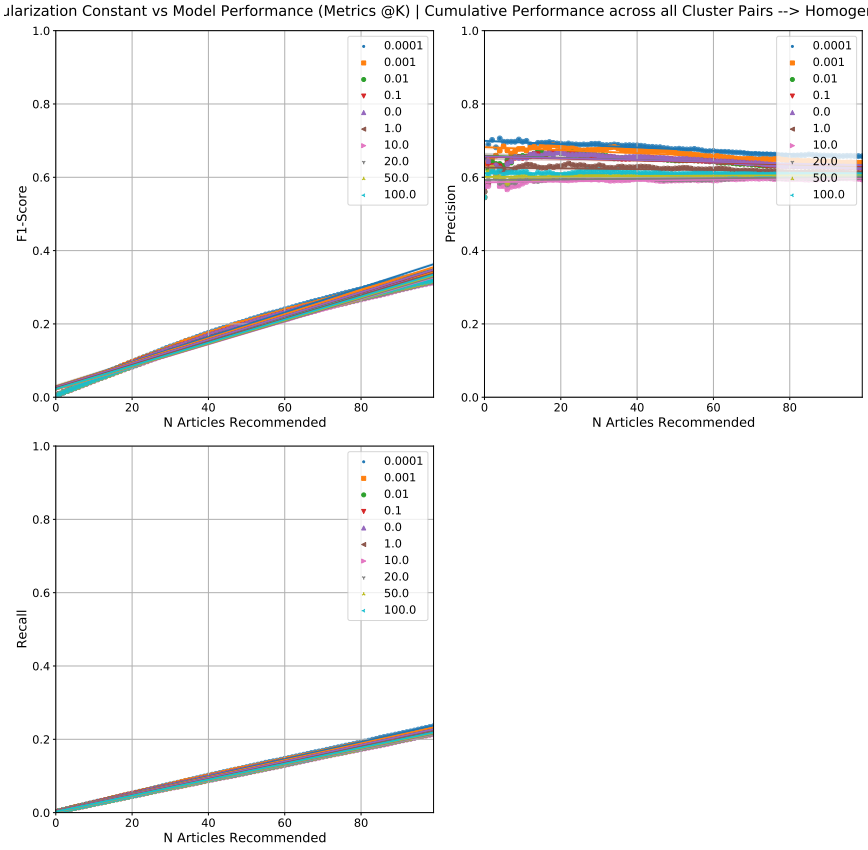
- Better performance when only one topic/cluster is involved compared to a cluster pair setting
- TF-IDF has the best performance here followed by BERT and then GLOVE

8 Baseline 4: Varying Regularization Strength to remove Spurious Correlations

8.1 TF-IDF

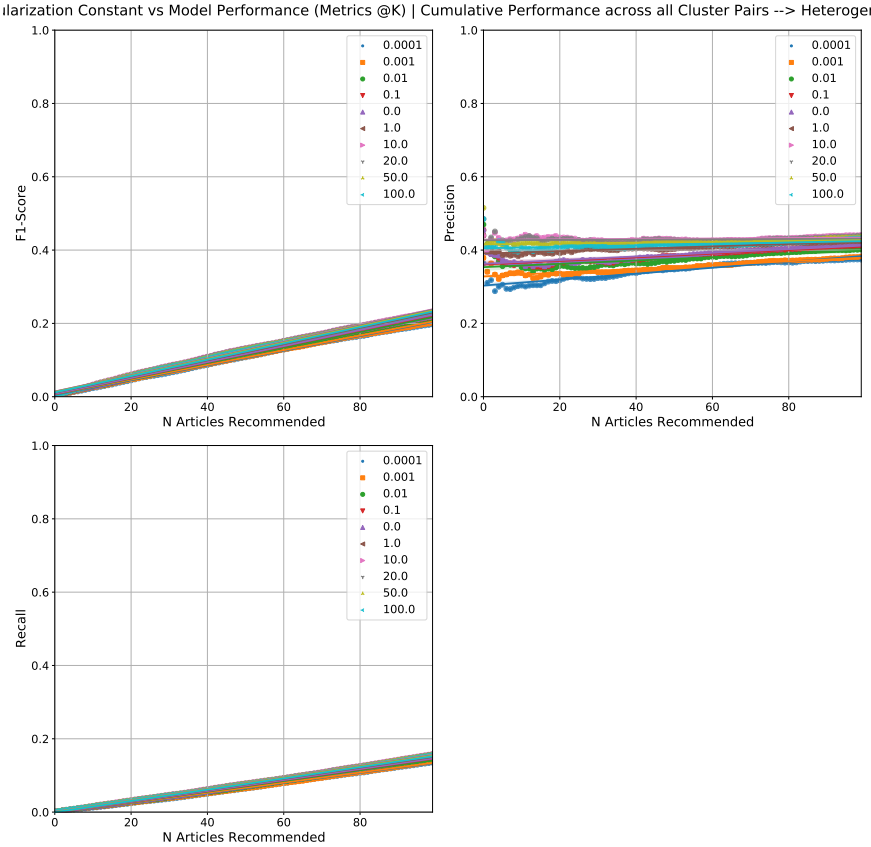
8.1.1 Homogeneous Users

For Homogeneous Users we see that a high regularization constant tends to hurt model performance, even not using a regularization constant (alpha =0) hurts precision compared to using a small regularization constant.



8.1.2 Heterogeneous Users

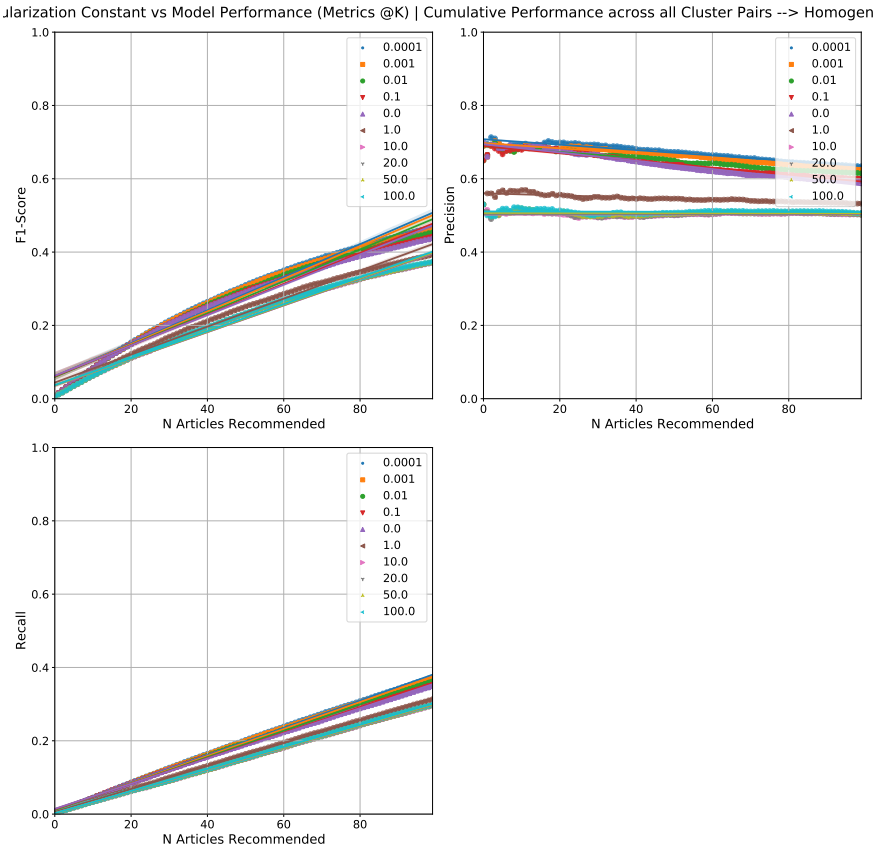
For Heterogeneous Users on the other hand, the greater the regularization constant the better the model performs (as it can be seen across all 3 metrics), so limiting spurious correlations definitely does help in this scenario as words that overlap across stances are demoted in importance.



8.2 Glove

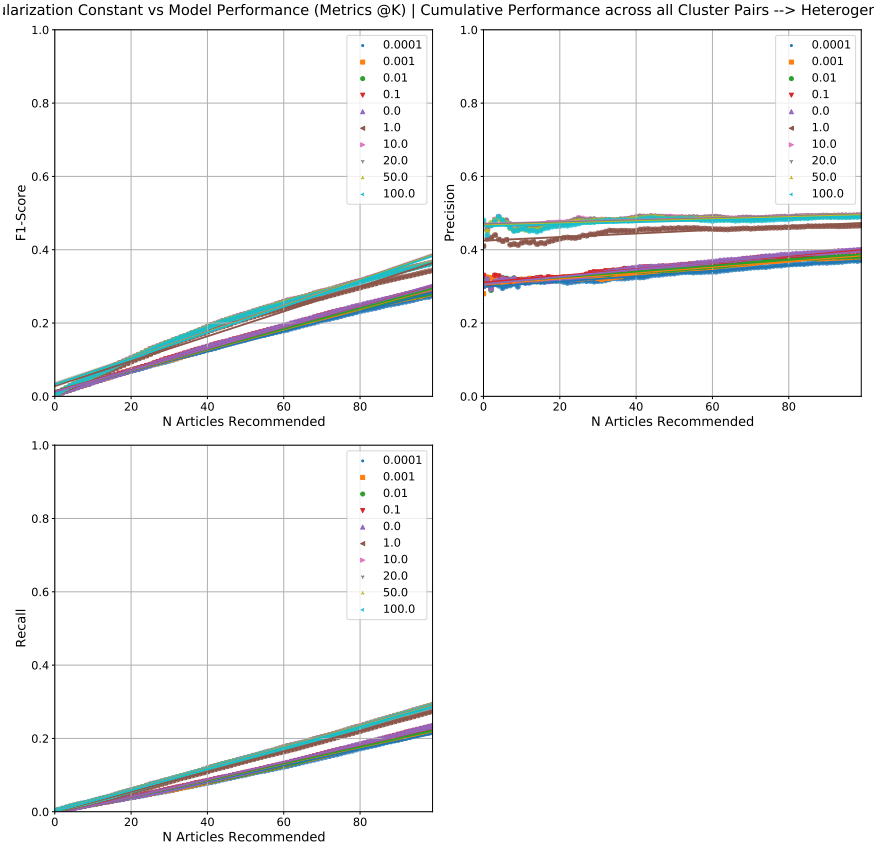
8.2.1 Homogeneous Users

For Homogeneous Users we see that a high regularization constant tends to keep the precision at a more constant rate compared to a smaller regularization constant but has lower values initially.



8.2.2 Heterogeneous Users

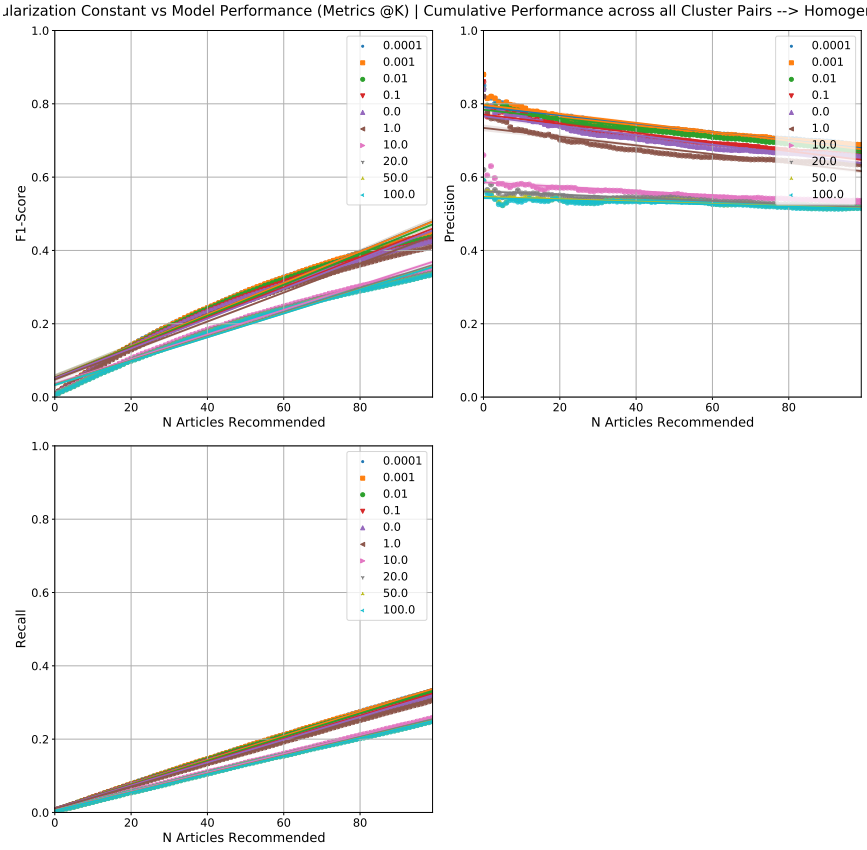
For Heterogeneous Users on the other hand, higher regularization constant values improves precision over lower regularization constant values and the precision seems to be increase at a more constant rate compared to lower values.



8.3 BERT

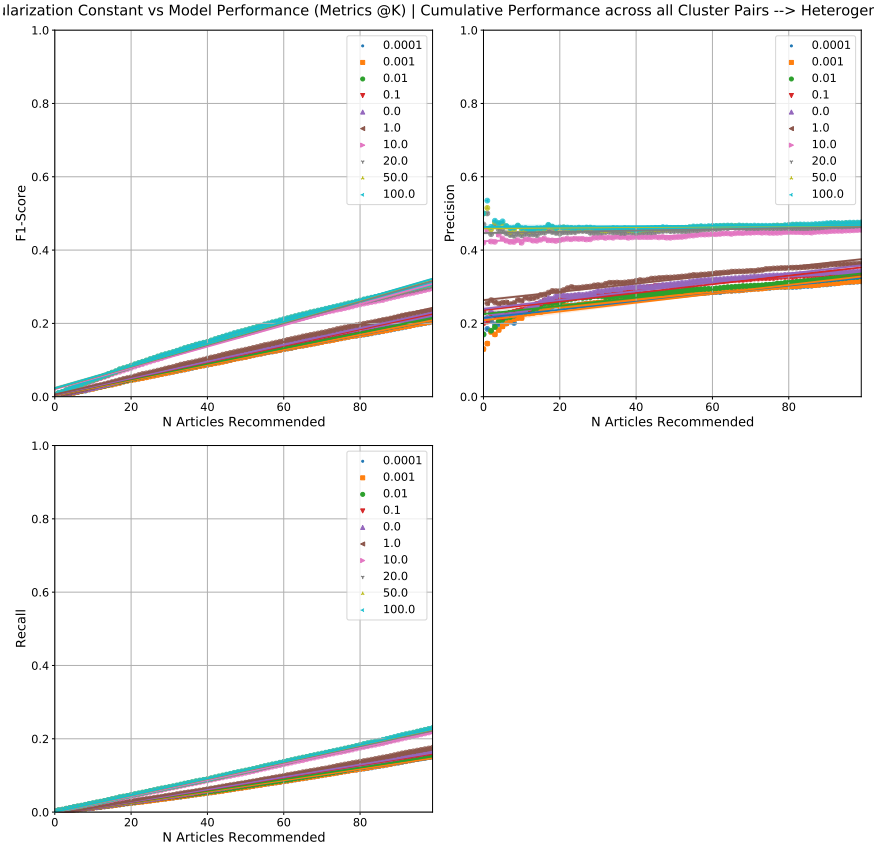
8.3.1 Homogeneous Users

For Homogeneous Users we see that lower regularization values have high precision compared to higher constant values and we also observe that the precision for the initial interactions are much higher compared to when using TFIDF and GLOVE representations



8.3.2 Heterogeneous Users

For Heterogeneous Users on the other hand, higher regularization constant values improves precision over lower regularization constant values and the precision seems to be increase at a more constant rate compared to lower values. We also observe that both GLOVE and TFIDF Representations tend to have a higher precision compared to BERT for lower regularization values.



8.4 Summary

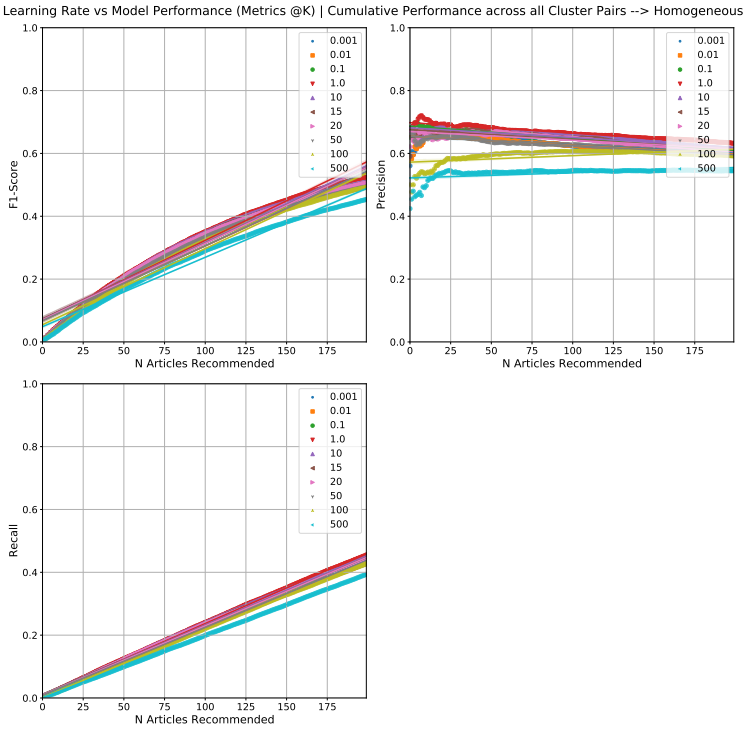
- TF-IDF is not affected greatly by using regularization constants for the Homogeneous users compared to GLOVE and BERT representations
- BERT representations have higher initial precision for Homogeneous users compared to TFIDF and GLOVE with low regularization constant values
- GLOVE representations with high regularization constant values seems to be performing the best for Heterogeneous Users, followed by BERT (with only high regularization values)

9 Baseline 5: Learning Rate vs Model Performance

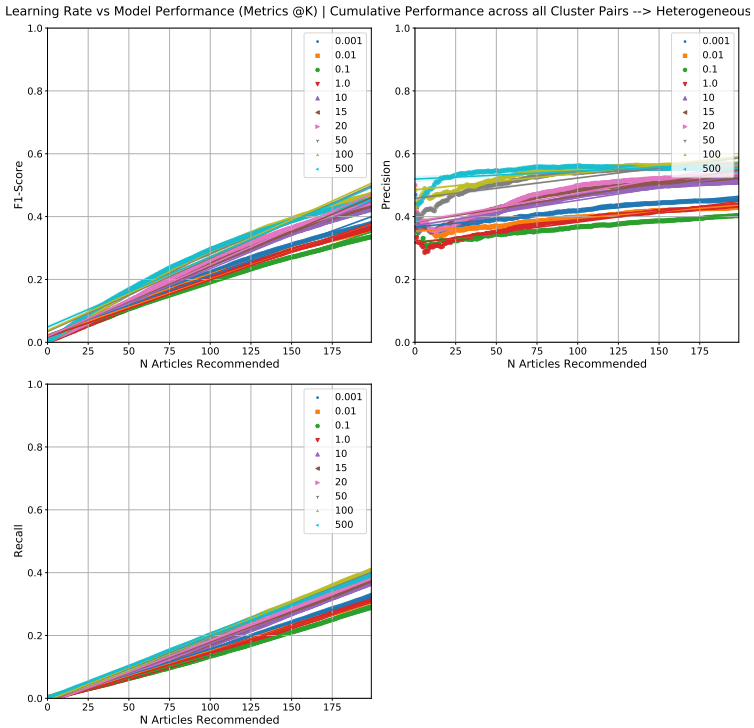
We want to measure the effect of using different learning rates to see which leads to better convergence when max-iterations are set to 1000.

9.1 TF-IDF

9.1.1 Homogeneous Users :

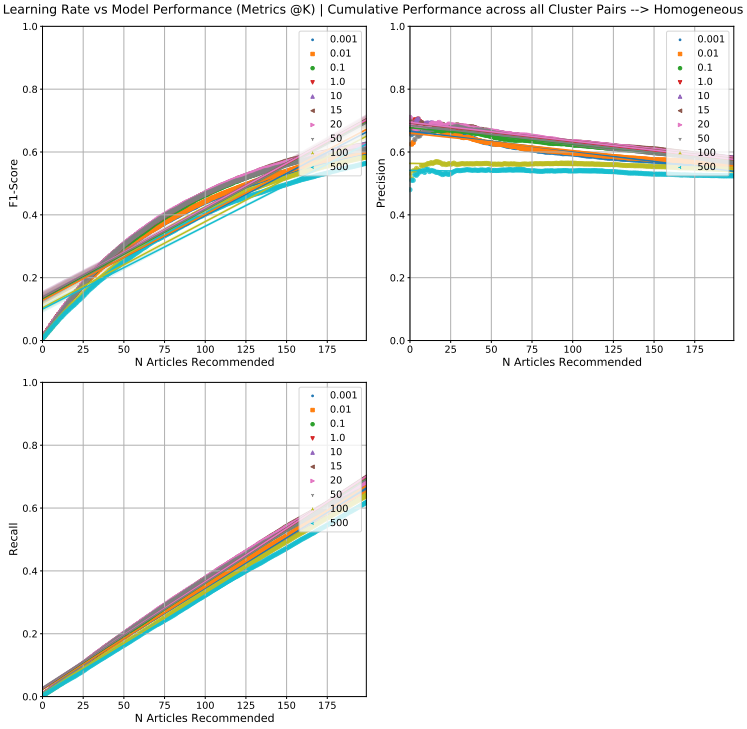


9.1.2 Heterogeneous Users :

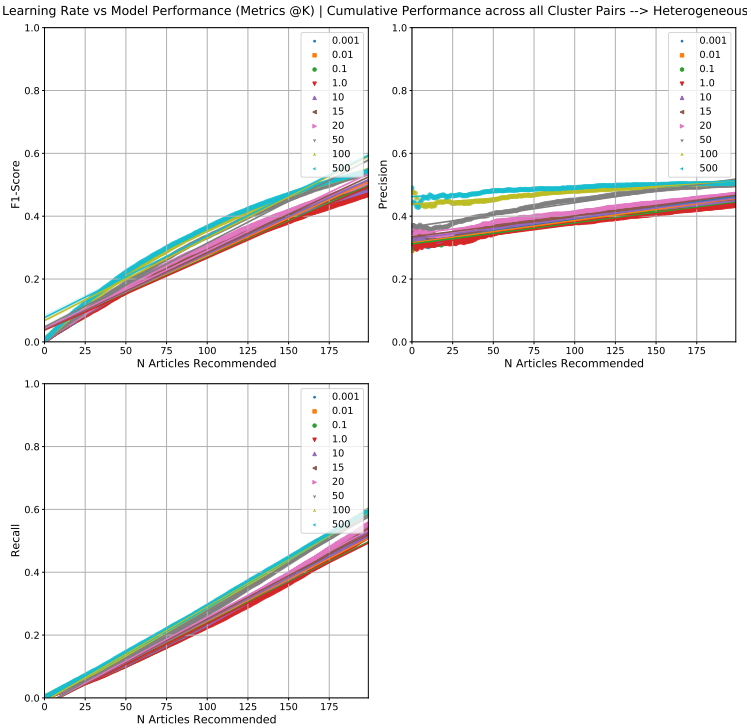


9.2 Glove

9.2.1 Homogeneous Users :

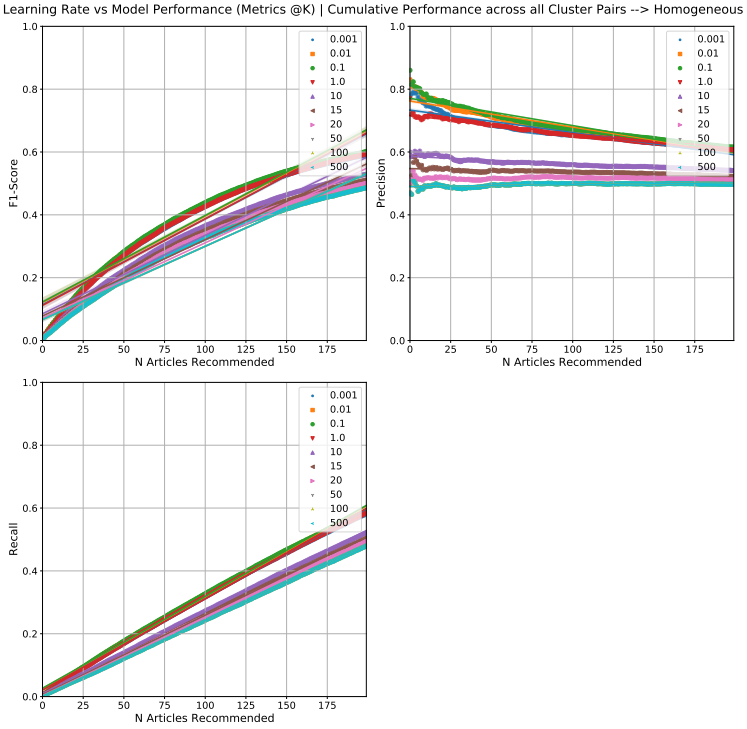


9.2.2 Heterogeneous Users :

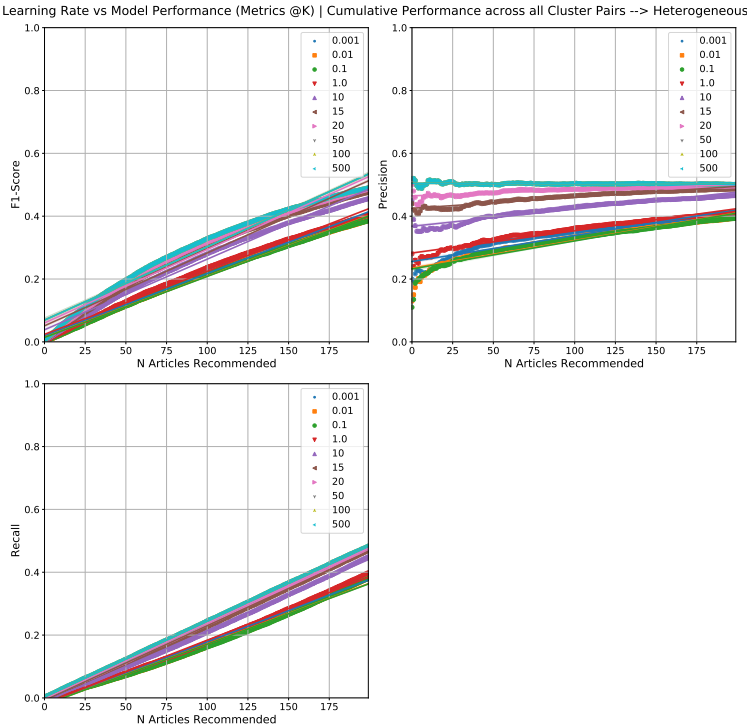


9.3 BERT

9.3.1 Homogeneous Users :



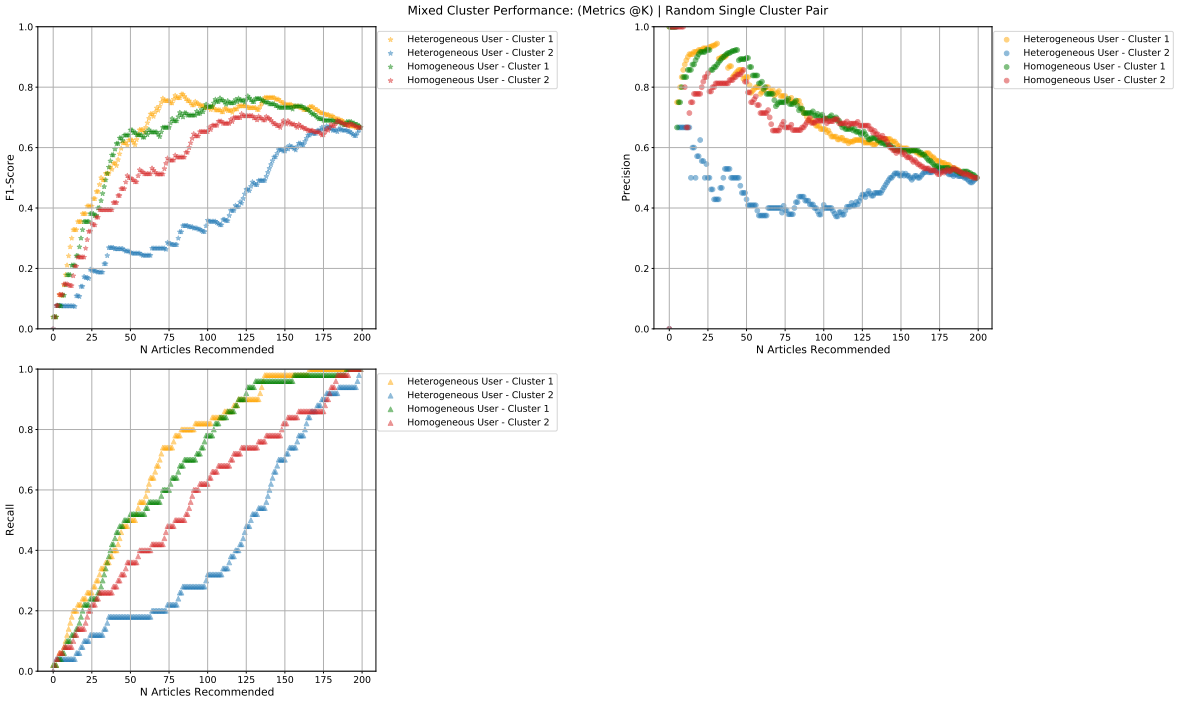
9.3.2 Heterogeneous Users :

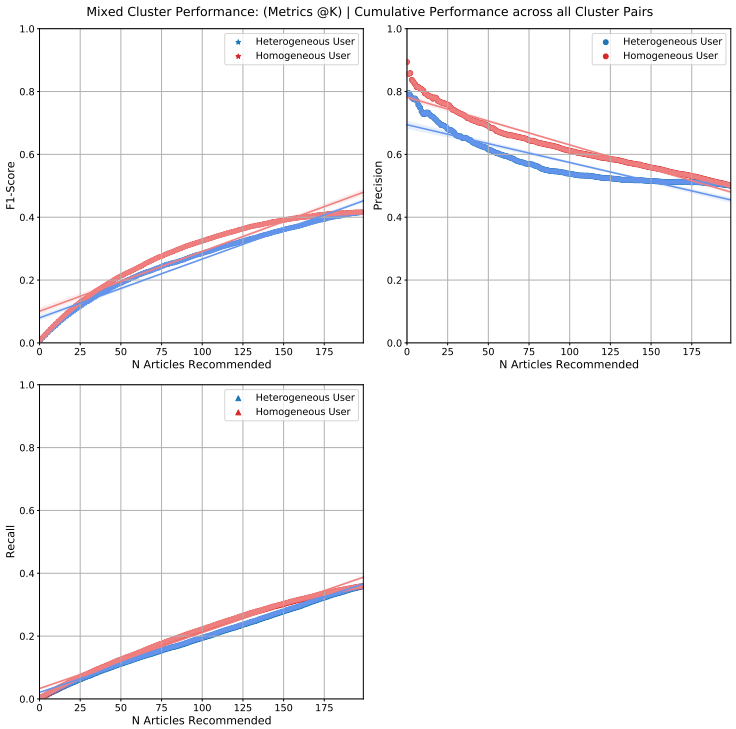


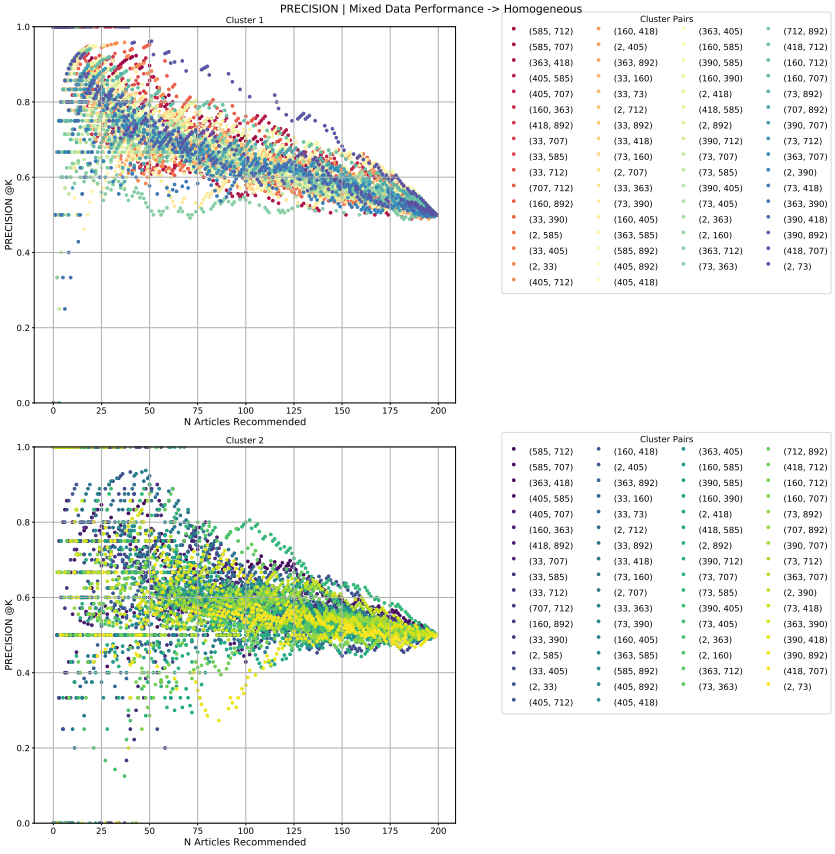
10 Baseline 6: Model Performance on mixed Set (Cluster 1 + Cluster 2)

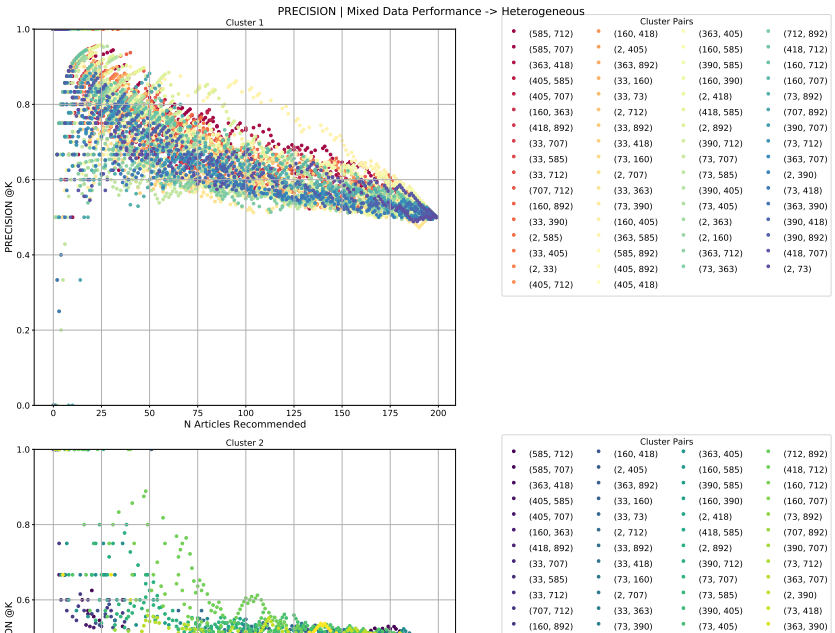
We want to measure how well the model would perform when an article from an unseen topic arrives along with articles from the original topic the classifier was trained on

10.1 TF-IDF

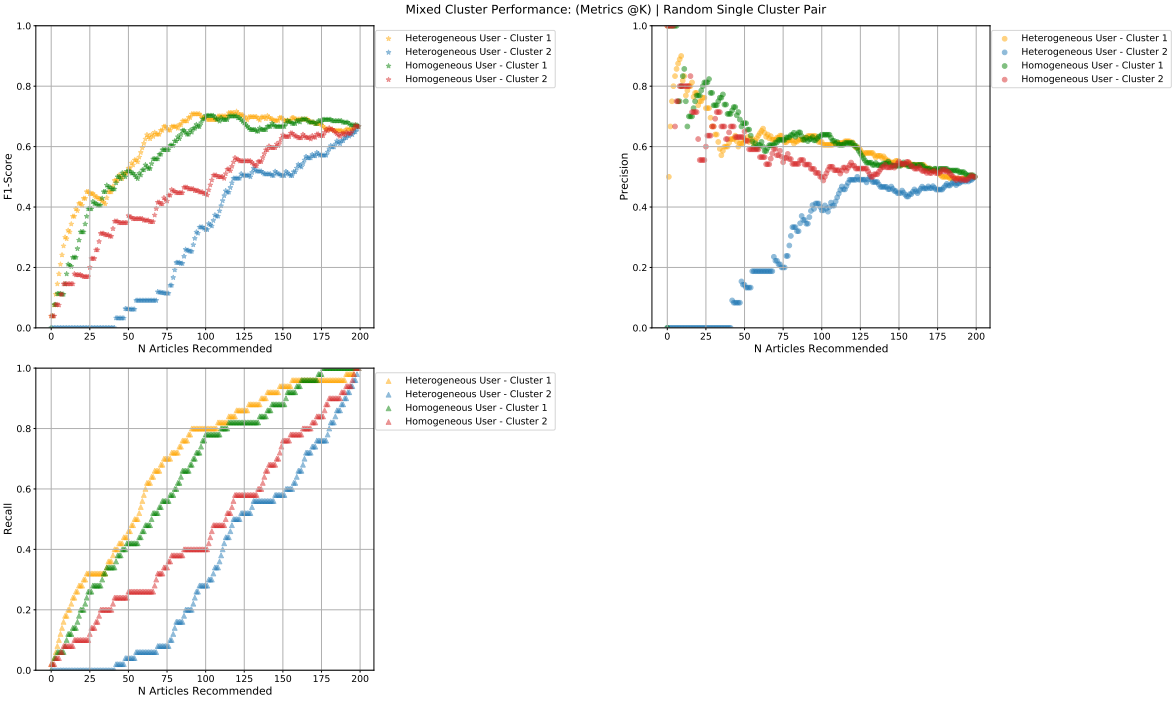


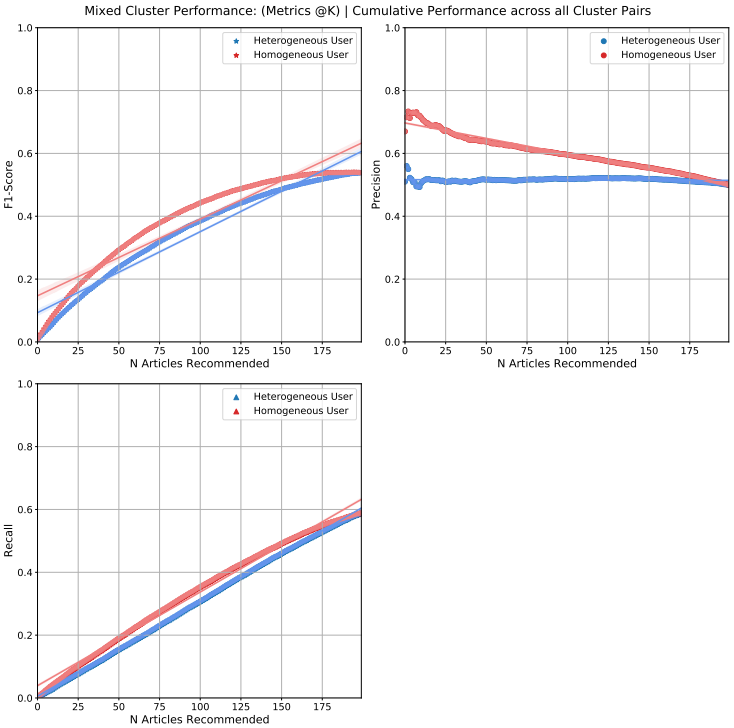



PRECISION | Mixed Data Performance -> Homogeneous
Cluster 1
Cluster 2
PRECISION @K
N Articles Recommended
Cluster Pairs
(585, 712)
(585, 707)
(363, 418)
(405, 585)
(405, 707)
(160, 363)
(418, 892)
(33, 707)
(33, 585)
(33, 712)
(707, 712)
(160, 892)
(33, 390)
(2, 585)
(33, 405)
(2, 33)
(405, 712)
(160, 418)
(2, 405)
(363, 892)
(33, 160)
(33, 73)
(2, 712)
(33, 892)
(33, 418)
(73, 160)
(2, 707)
(33, 363)
(73, 390)
(160, 405)
(363, 585)
(585, 892)
(405, 892)
(405, 418)
(363, 405)
(160, 585)
(390, 585)
(160, 390)
(2, 418)
(418, 585)
(2, 892)
(390, 712)
(73, 707)
(73, 585)
(390, 405)
(73, 405)
(2, 363)
(2, 160)
(363, 712)
(73, 363)
(712, 892)
(418, 712)
(160, 712)
(160, 707)
(73, 892)
(707, 892)
(390, 707)
(73, 712)
(363, 707)
(2, 390)
(73, 418)
(363, 390)
(390, 418)
(390, 892)
(418, 707)
(2, 73)

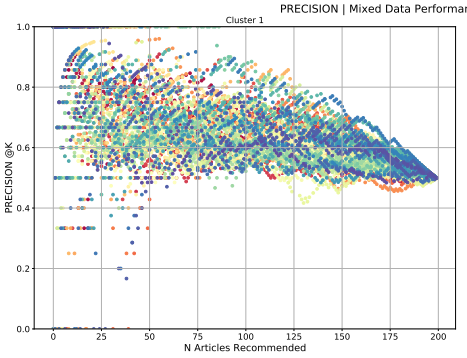

PRECISION | Mixed Data Performance -> Heterogeneous
Cluster 1
Cluster 2
PRECISION @K
N Articles Recommended
Cluster Pairs
(585, 712)
(585, 707)
(363, 418)
(405, 585)
(405, 707)
(160, 363)
(418, 892)
(33, 707)
(33, 585)
(33, 712)
(707, 712)
(160, 892)
(33, 390)
(2, 585)
(33, 405)
(2, 33)
(405, 712)
(160, 418)
(2, 405)
(363, 892)
(33, 160)
(33, 73)
(2, 712)
(33, 892)
(33, 418)
(73, 160)
(2, 707)
(33, 363)
(73, 390)
(160, 405)
(363, 585)
(585, 892)
(405, 892)
(405, 418)
(363, 405)
(160, 585)
(390, 585)
(160, 390)
(2, 418)
(418, 585)
(2, 892)
(390, 712)
(73, 707)
(73, 585)
(390, 405)
(73, 405)
(2, 363)
(2, 160)
(363, 712)
(73, 363)
(712, 892)
(418, 712)
(160, 712)
(160, 707)
(73, 892)
(707, 892)
(390, 707)
(73, 712)
(363, 707)
(2, 390)
(73, 418)
(363, 390)
(390, 418)
(390, 892)
(418, 707)
(2, 73)

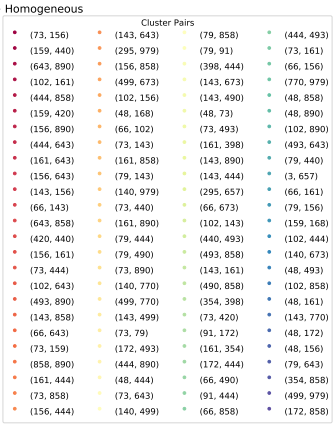
10.2 Glove

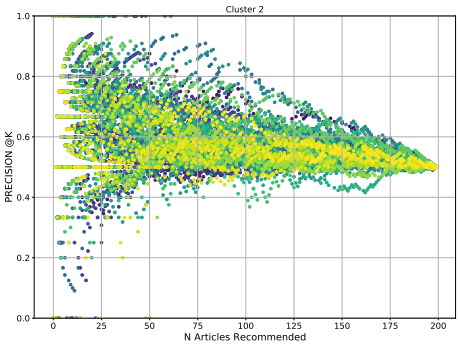

Mixed Cluster Performance: (Metrics @K) | Random Single Cluster Pair
F1-Score
Precision
Recall
N Articles Recommended
Heterogeneous User - Cluster 1
Heterogeneous User - Cluster 2
Homogeneous User - Cluster 1
Homogeneous User - Cluster 2

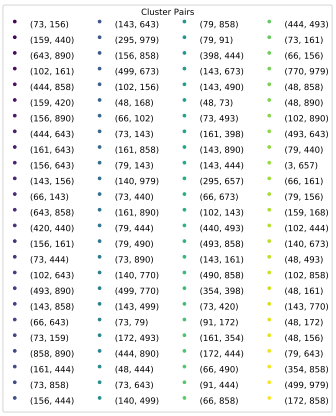

Mixed Cluster Performance: (Metrics @K) | Cumulative Performance across all Cluster Pairs
F1-Score
Precision
Recall
N Articles Recommended
Heterogeneous User
Homogeneous User

PRECISION | Mixed Data Performance -> Homogeneous

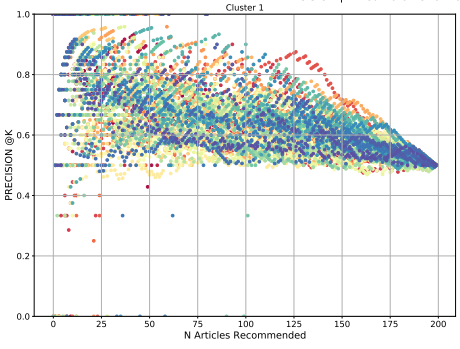


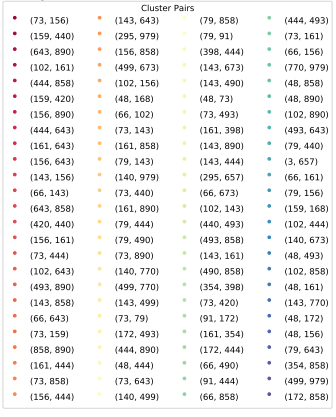


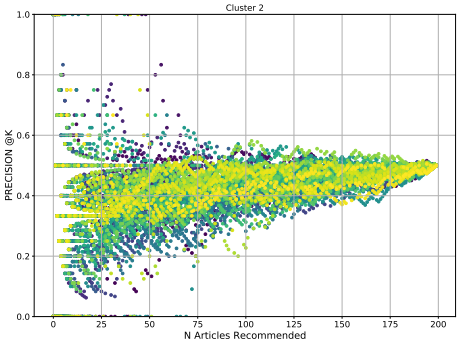


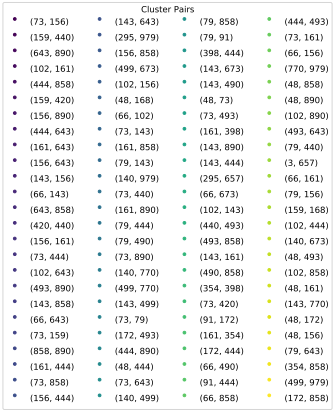


PRECISION | Mixed Data Performance -> Heterogeneous

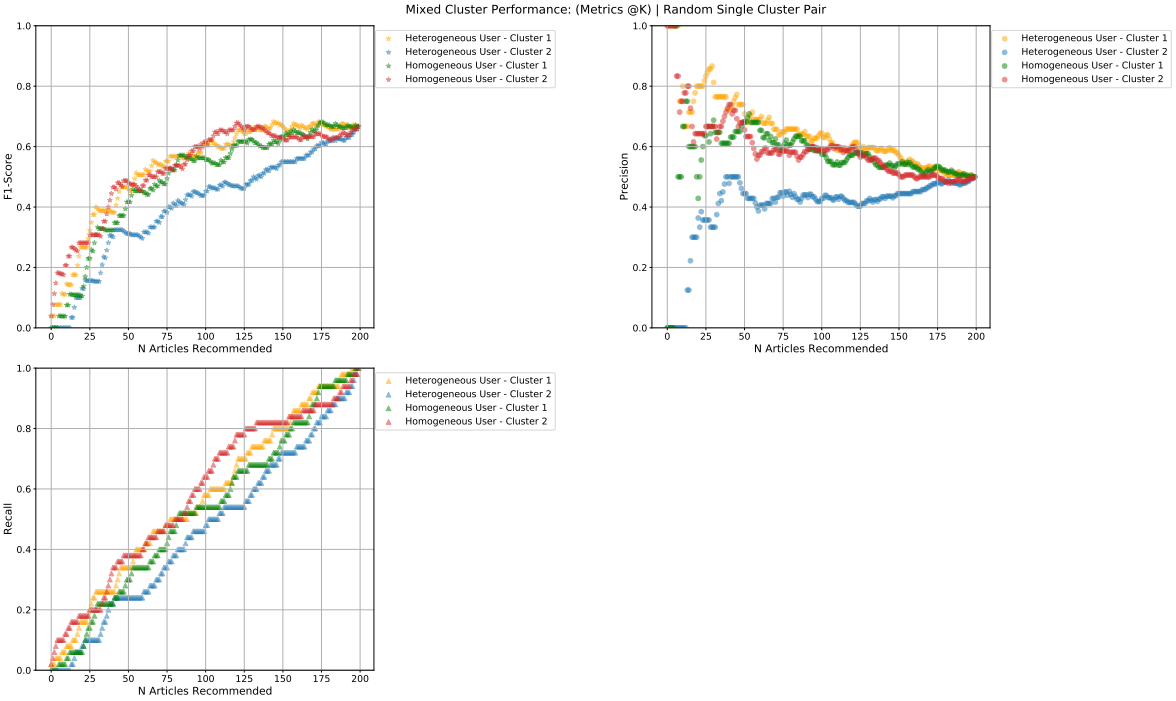


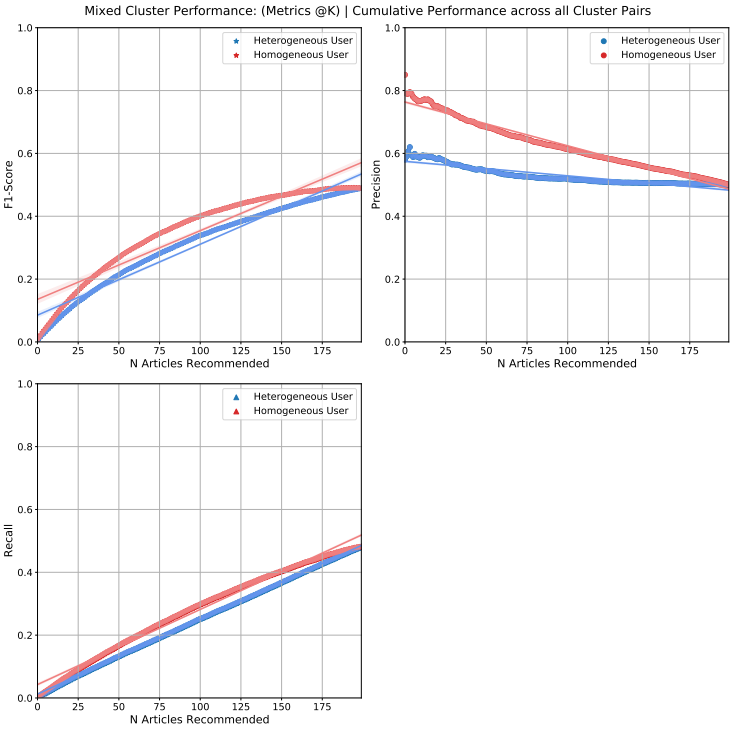




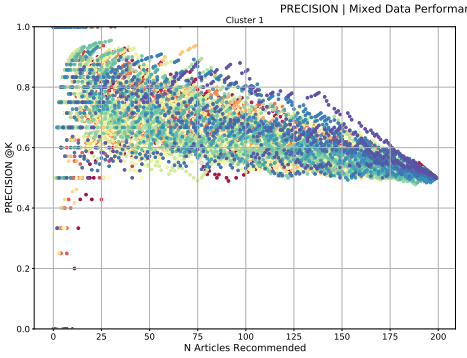


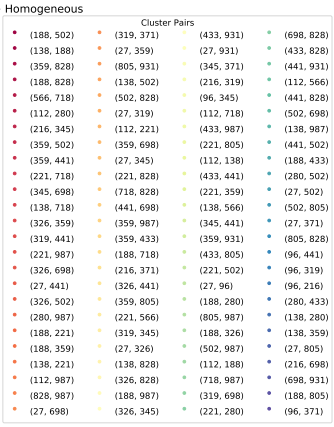
10.3 BERT

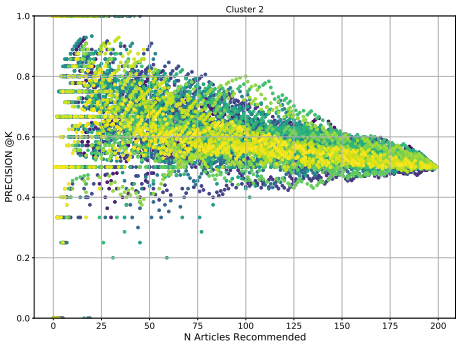

Mixed Cluster Performance: (Metrics @K) | Random Single Cluster Pair
F1-Score
Precision
Recall
N Articles Recommended
Heterogeneous User - Cluster 1
Heterogeneous User - Cluster 2
Homogeneous User - Cluster 1
Homogeneous User - Cluster 2

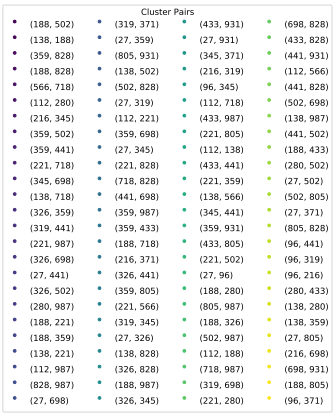

Mixed Cluster Performance: (Metrics @K) | Cumulative Performance across all Cluster Pairs
F1-Score
Precision
Recall
N Articles Recommended
Heterogeneous User
Homogeneous User

PRECISION | Mixed Data Performance -> Homogeneous

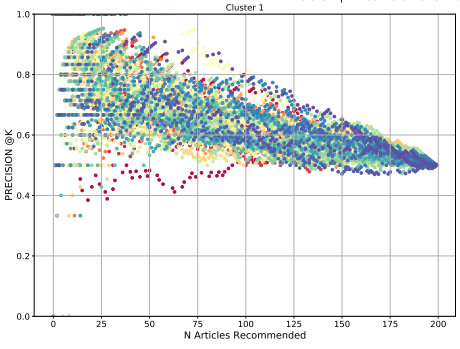


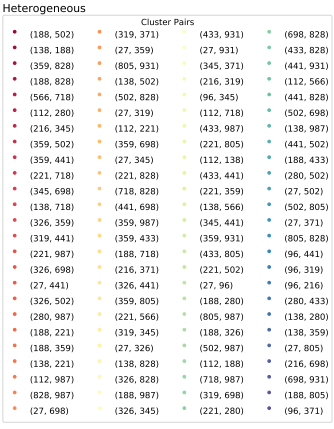


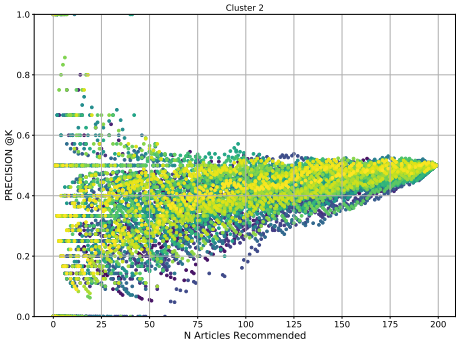


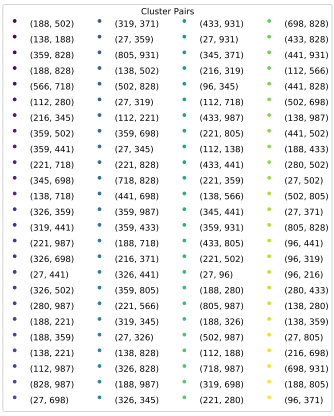


PRECISION | Mixed Data Performance -> Heterogeneous





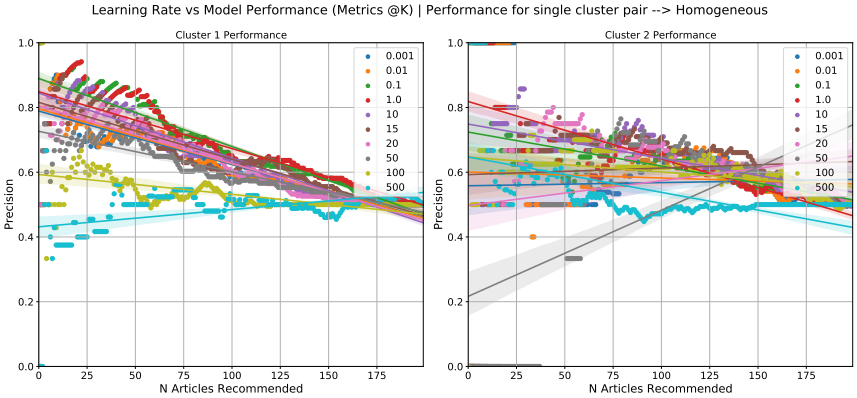


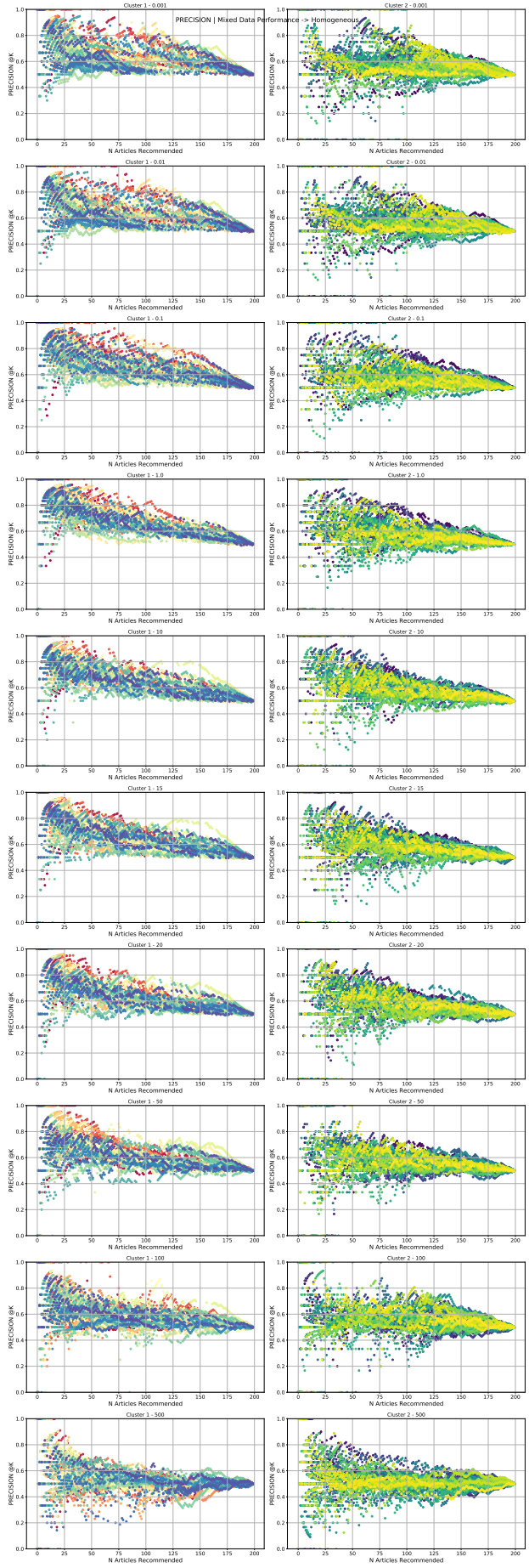


11 Baseline 7: Learning Rate Variation for Mixed Cluster

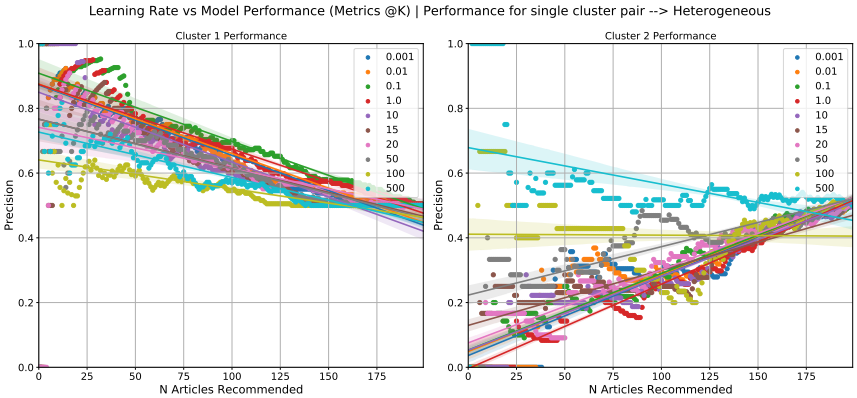
11.1 TF-IDF

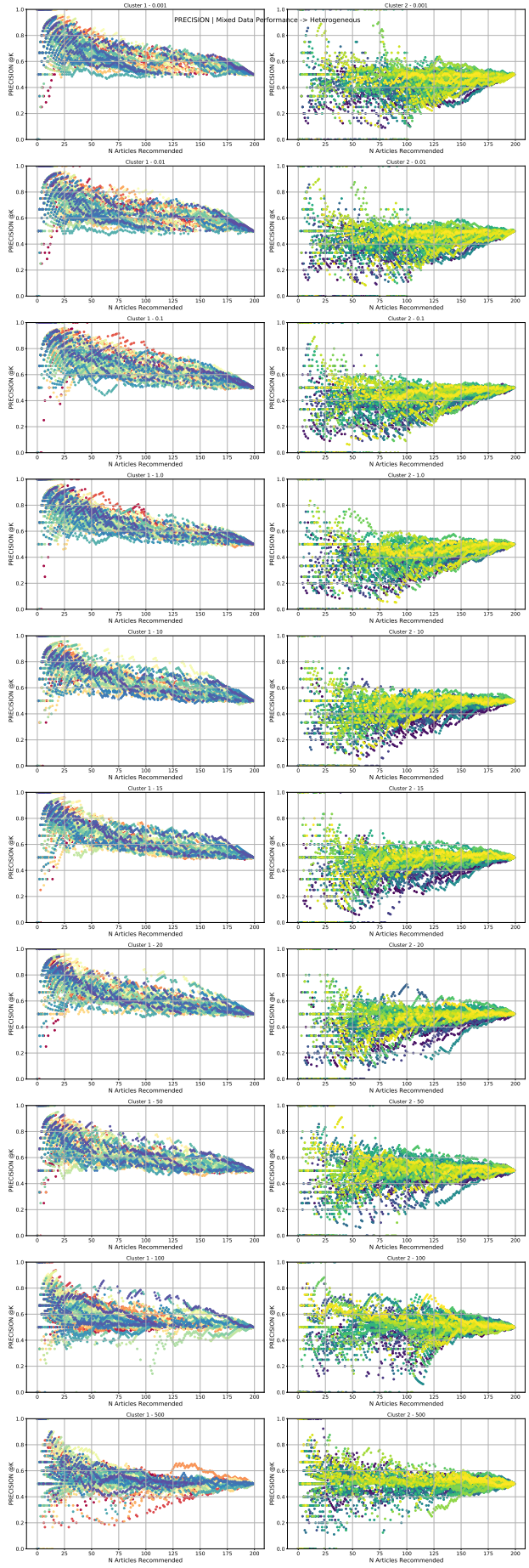
11.1.1 Homogeneous




PRECISION | Mixed Data Performance -> Homogeneous
Cluster 1 - 0.001
Cluster 2 - 0.001
Cluster 1 - 0.01
Cluster 2 - 0.01
Cluster 1 - 0.1
Cluster 2 - 0.1
Cluster 1 - 1.0
Cluster 2 - 1.0
Cluster 1 - 10
Cluster 2 - 10
Cluster 1 - 15
Cluster 2 - 15
Cluster 1 - 20
Cluster 2 - 20
Cluster 1 - 50
Cluster 2 - 50
Cluster 1 - 100
Cluster 2 - 100
Cluster 1 - 500
Cluster 2 - 500
PRECISION @K
N Articles Recommended

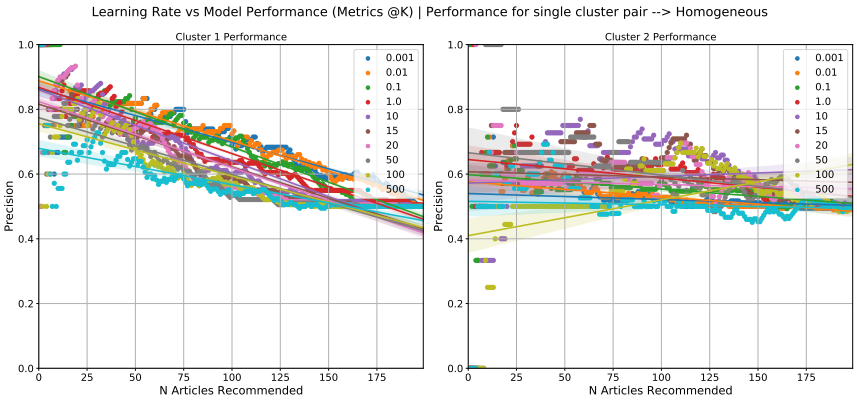
11.1.2 Heterogeneous

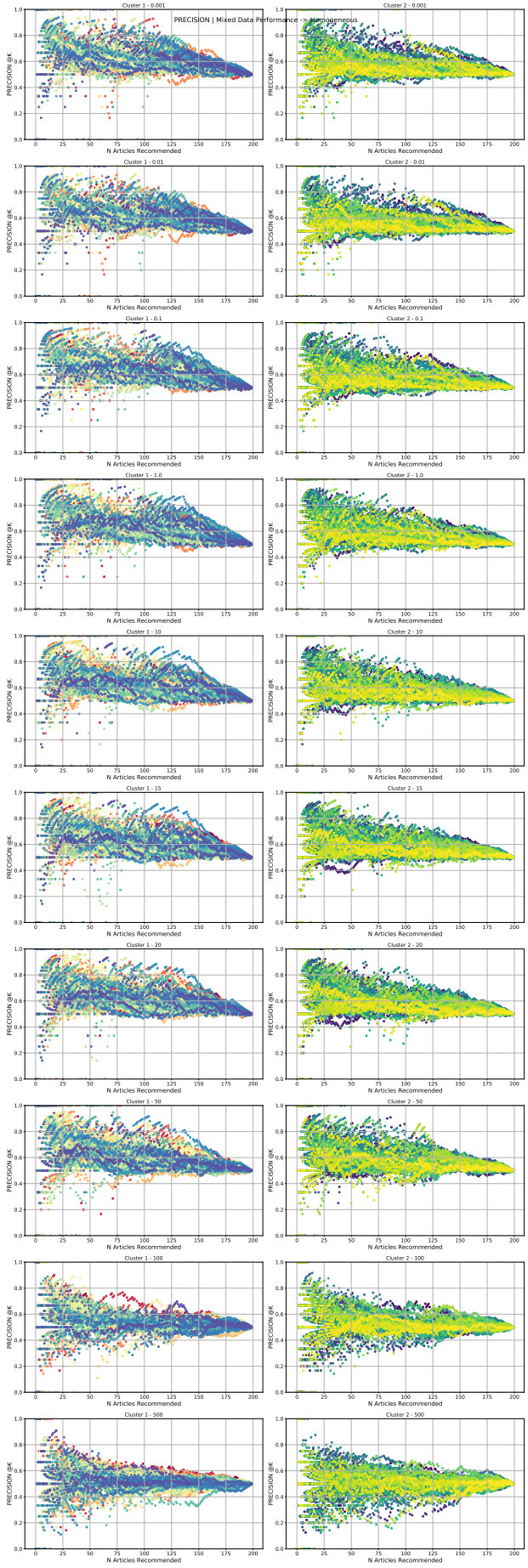



PRECISION | Mixed Data Performance -> Heterogeneous
Cluster 1 - 0.001
Cluster 2 - 0.001
Cluster 1 - 0.01
Cluster 2 - 0.01
Cluster 1 - 0.1
Cluster 2 - 0.1
Cluster 1 - 1.0
Cluster 2 - 1.0
Cluster 1 - 10
Cluster 2 - 10
Cluster 1 - 15
Cluster 2 - 15
Cluster 1 - 20
Cluster 2 - 20
Cluster 1 - 50
Cluster 2 - 50
Cluster 1 - 100
Cluster 2 - 100
Cluster 1 - 500
Cluster 2 - 500
PRECISION @K
N Articles Recommended

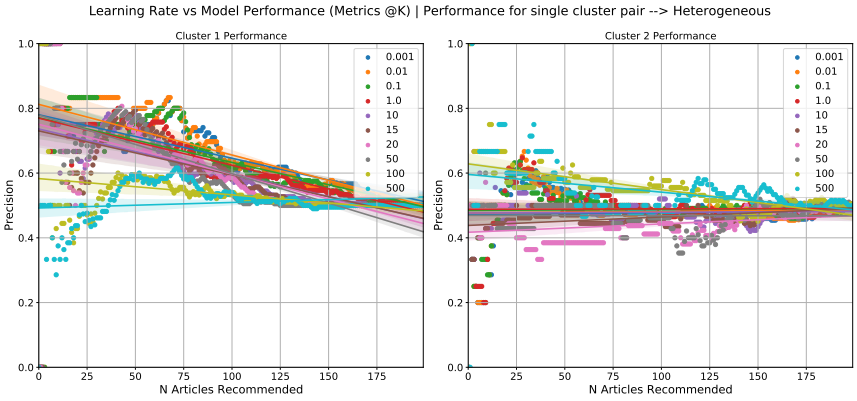
11.2 Glove

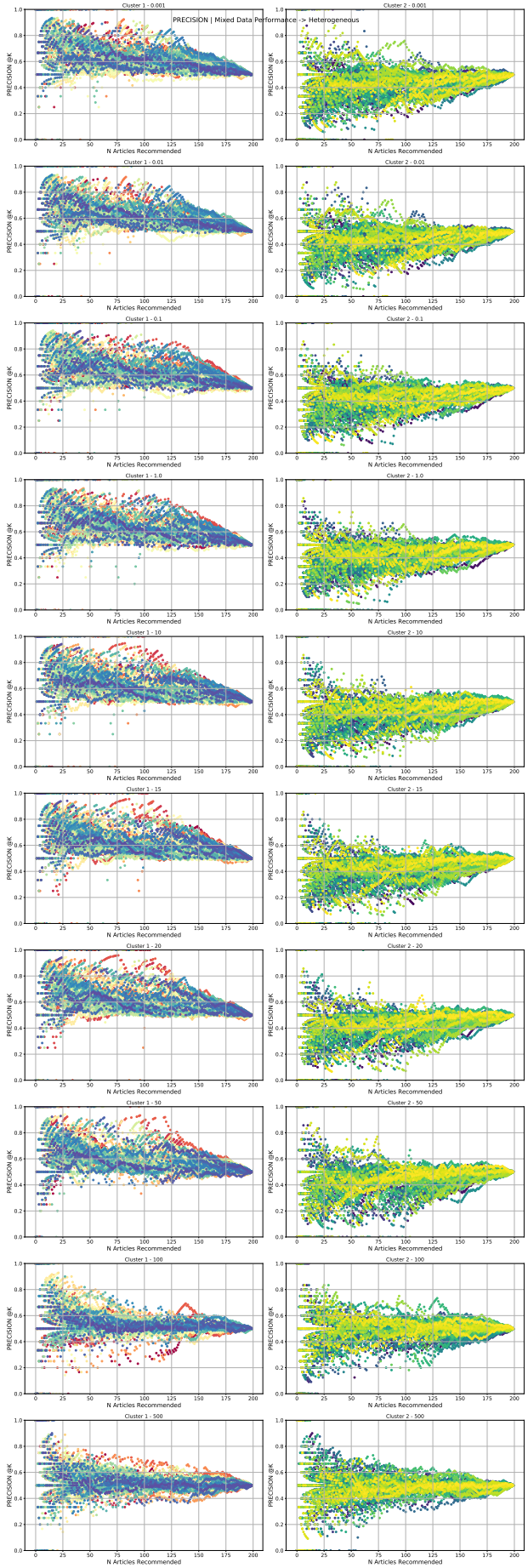
11.2.1 Homogeneous




PRECISION | Mixed Data Performance -> Homogeneous
Cluster 1 - 0.001
Cluster 2 - 0.001
Cluster 1 - 0.01
Cluster 2 - 0.01
Cluster 1 - 0.1
Cluster 2 - 0.1
Cluster 1 - 1.0
Cluster 2 - 1.0
Cluster 1 - 10
Cluster 2 - 10
Cluster 1 - 15
Cluster 2 - 15
Cluster 1 - 20
Cluster 2 - 20
Cluster 1 - 50
Cluster 2 - 50
Cluster 1 - 100
Cluster 2 - 100
Cluster 1 - 500
Cluster 2 - 500
PRECISION @K
N Articles Recommended

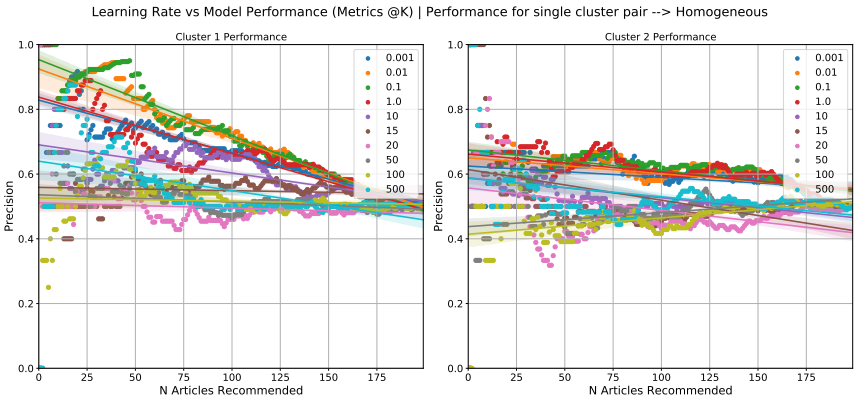
11.2.2 Heterogeneous

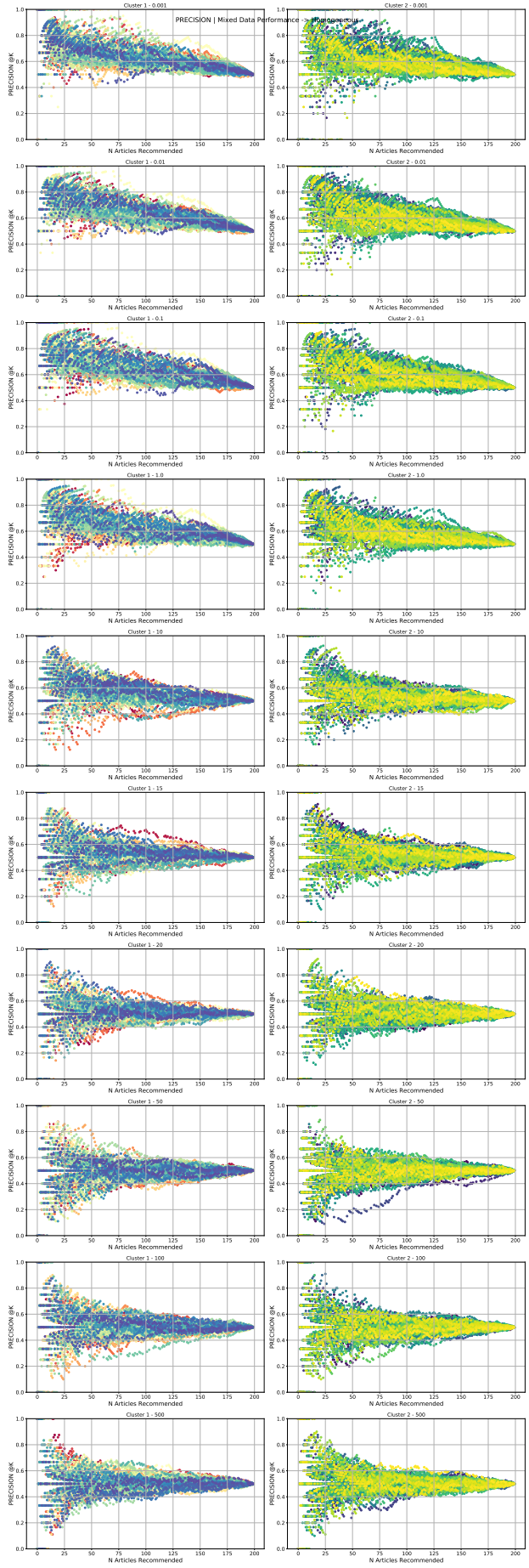



PRECISION | Mixed Data Performance -> Heterogeneous
Cluster 1 - 0.001
Cluster 2 - 0.001
Cluster 1 - 0.01
Cluster 2 - 0.01
Cluster 1 - 0.1
Cluster 2 - 0.1
Cluster 1 - 1.0
Cluster 2 - 1.0
Cluster 1 - 10
Cluster 2 - 10
Cluster 1 - 15
Cluster 2 - 15
Cluster 1 - 20
Cluster 2 - 20
Cluster 1 - 50
Cluster 2 - 50
Cluster 1 - 100
Cluster 2 - 100
Cluster 1 - 500
Cluster 2 - 500
PRECISION @K
N Articles Recommended

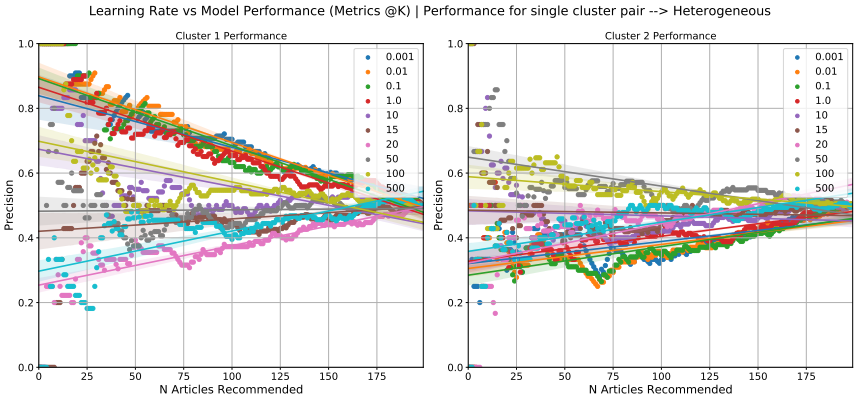
11.3 BERT

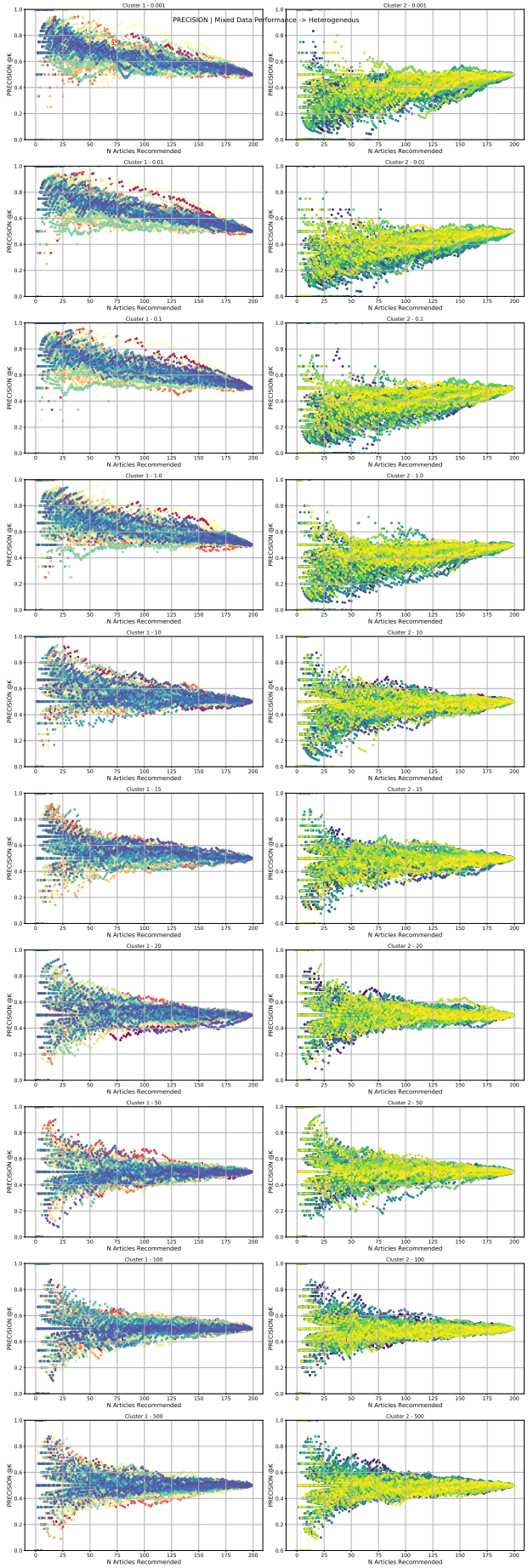
11.3.1 Homogeneous




PRECISION | Mixed Data Performance -> Heterogeneous
Cluster 1 - 0.001
Cluster 2 - 0.001
Cluster 1 - 0.01
Cluster 2 - 0.01
Cluster 1 - 0.1
Cluster 2 - 0.1
Cluster 1 - 1.0
Cluster 2 - 1.0
Cluster 1 - 10
Cluster 2 - 10
Cluster 1 - 15
Cluster 2 - 15
Cluster 1 - 20
Cluster 2 - 20
Cluster 1 - 50
Cluster 2 - 50
Cluster 1 - 100
Cluster 2 - 100
Cluster 1 - 500
Cluster 2 - 500
PRECISION @K
N Articles Recommended

11.3.2 Heterogeneous




PRECISION | Mixed Data Performance -> Heterogeneous
Cluster 1 - 0.001
Cluster 2 - 0.001
Cluster 1 - 0.01
Cluster 2 - 0.01
Cluster 1 - 0.1
Cluster 2 - 0.1
Cluster 1 - 1.0
Cluster 2 - 1.0
Cluster 1 - 10
Cluster 2 - 10
Cluster 1 - 15
Cluster 2 - 15
Cluster 1 - 20
Cluster 2 - 20
Cluster 1 - 50
Cluster 2 - 50
Cluster 1 - 100
Cluster 2 - 100
Cluster 1 - 500
Cluster 2 - 500
PRECISION @K
N Articles Recommended