11791 Homework 4 Report

I started evaluating the performance among different models on two metrics by doing basic statistical analysis. Results are shown in the following table. As can be observed from the table, the baseline model has the highest mean, median and mode value for both metrics.

metric	model	mean	median	mode	min	max
ROUGE-2	Baseline	**0.4165538**	**0.33333**	**1 (count=12)**	0	1
ROUGE-2	Baseline+Fusion	0.3029308	0.218255	0 (count=17)	0	1
ROUGE-2	Baseline+Ordering	0.4124338	0.318015	0 (count=19)	0	1
ROUGE-2	Baseline+Ordering+Fusion	0.2929442	0.207615	0 (count=17)	0	1
ROUGE-SU4	Baseline	**0.4165538**	**0.33333**	**1 (count=12)**	0	1
ROUGE-SU4	Baseline+Fusion	0.3029308	0.218255	0 (count=17)	0	1
ROUGE-SU4	Baseline+Ordering	0.4124338	0.318015	0 (count=19)	0	1
ROUGE-SU4	Baseline+Ordering+Fusion	0.2929442	0.207615	0 (count=17)	0	1

I then calculated p-values with the T-Test, Wilcoxon test and KS test on each model and metric. The following table displays the results using red to indicate p-values >= 0.05 and green to indicate p-values < 0.05. The results show that improved models are not necessarily better than the baseline model.

Specifically, the three models (fusion, ordering, ordering+fusion) are not significantly better than the baseline model in all three tests. In both metrics, the improvement of the ordering model is not significant as measured by all three tests. The fusion and ordering+fusion model are significantly better than the baseline model in T-test and Wilcoxon test but not significant in the KS test. When comparing the fusion and the ordering model, the ordering model is significantly better than the baseline fusion model in the T-test and Wilcoxon test. The ordering+fusion model is not significantly better than the fusion model in all metrics and significance measures. The ordering+fusion model is significantly better in all three significance measures but is not marked as better in the mean diff metrics.

metric	model	mean diff	P(T test)	P(wilcoxon test)	P(ks test)
ROUGE-2	Baseline & Fusion	-0.11362300000000009	0.010449188538310899	0.006592269408060 4635	0.139975180163 85846
ROUGE-2	Baseline & Ordering	-0.004119999999999957	0.934514036945895 97	0.621489948888700157	0.193041651924 69011

ROUGE-2	Baseline & Ordering+Fusion	-0.123609600000001	0.0051050560123188041	0.0044566094455675952	0.069092434889398091
ROUGE-2	Fusion & Ordering	0.109503000000000013	0.018701749637786994	0.0056470445023683548	0.069092434889398091
ROUGE-2	Fusion & Ordering+Fusion	-0.009986600000000121	0.79877547262841786	0.75334568395674884	0.99999689889731391
ROUGE-2	Ordering & Ordering+Fusion	-0.119489600000000014	0.0099302146504130131	0.00098369909761899317	0.031376652153072448
ROUGE-SU4	Baseline & Fusion	-0.113623000000000009	0.010449188538310899	0.0065922694080604635	0.13997518016385846
ROUGE-SU4	Baseline & Ordering	-0.004119999999999957	0.93451403694589597	0.62148994888870157	0.19304165192469011
ROUGE-SU4	Baseline & Ordering+Fusion	-0.123609600000000001	0.0051050560123188041	0.0044566094455675952	0.069092434889398091
ROUGE-SU4	Fusion & Ordering	0.109503000000000013	0.018701749637786994	0.0056470445023683548	0.069092434889398091
ROUGE-SU4	Fusion & Ordering+Fusion	-0.009986600000000121	0.79877547262841786	0.75334568395674884	0.99999689889731391
ROUGE-SU4	Ordering & Ordering+Fusion	-0.119489600000000014	0.0099302146504130131	0.00098369909761899317	0.031376652153072448

For the implementation, I basically followed the starter code but refactored the way to fill in the table. I used the starter code in writeOutput function as a new function called `init_table` that fills in basic headers, row names, etc. I added a function called `stat_analysis` that calculates the basic statistical metrics and stores the values as class members. The function `fill_results` evaluates on the four metrics (mean diff, ttest, wilcoxon test and ks test) and fills in the table. To ease the process of filling the table, I created another function called `cross_evaluate` that does cross evaluations on each pair of the models and fills the table with respective results. The final write_output function just writes the whole table into the file. I simply used the existing packages in Scipy to do the p-value tests.