Shovel-Ready Competition - Unified Metric

Ralph Abboud

23 January 2024

This memo proposes a holistic new metric, built based on the original Feedback Prize 1.0 (FP1) metric, for the Shovel Ready competition anticipated to take place in February 2024. The proposed metric consists of a series of components, namely: (i) Overlap removal as pre-processing, (ii) an enhanced segment scoring metric which is sensitive to overlap quality with ground truth segments and to proximity of segment effectiveness labels, (iii) a bias-aware aggregation mechanism to penalize models biased against select demographics, and finally (iv) an efficiency-based boosting mechanism to encourage competitive, but more efficient, solutions in the final competition leaderboard.

1 Overlap Removal During Pre-processing

In the Feedback Prize competition series, a common undesirable behavior observed in winning models was that several predicted segments overlapped, i.e., two or more predicted segments shared a set of words. This not only goes against the principle of the segmentation task set in the competition, but also affects the validity of the scoring returned by the original metric. Indeed, prediction overlaps can lead to unjustifiably improved scores[1], particularly as a prediction only needs an overlap of 50% or more with the ground truth to count as a positive.

In light of these findings, we propose a new pre-processing step that applies on all submitted predictions to eliminate overlaps. The pre-processing proceeds as follows:

1. First, all predicted segments for a given essay are sorted based on their starting position. In other words, we sort segments in order of their appearance in the essay.

2. Second, we iterate sequentially over segments. While doing so, we accumulate all observed words (also considering their position) prior to the currently observed segment.

[1] In fact, by eliminating overlaps using the method described in this section, we found that benchmarked model scores dropped by roughly 0.02, which is sufficient to substantially change the final competition ranking.

3. If the current segment includes also used words, then this segments overlaps with a prior segment. Thus, we eliminate the used words from the current segment and return the resulting segment. Otherwise, the segment is returned with no changes.

4. If, as a result of used word removal, the resulting segment is no longer valid (contains 1 word or less, or no longer consists of a contiguous set of words), then the segment is discarded.

Example. Consider an essay consisting of 100 words, for which 6 segments have been predicted:

1. Segment A: From word 50 to 64.
2. Segment B: From word 10 to 29.
3. Segment C: From word 25 to 44.
4. Segment D: From word 90 to 99.
5. Segment E: From word 80 to 94.
6. Segment F: From word 45 to 69.

First, we sort the segments in order of their appearance, which leads to the order B, C, F, A, E, D. Then, we process the segments individually:

1. Segment B is the first segment, so does not overlap with previously used words. Thus, it is not affected. Words 10 to 29 are added to the set of used words.
2. Segment C overlaps with the used words set on words 25 through 29. Thus the resulting segment C' will only include words 30 through 44. The set of used words now includes words 10 through 44.
3. Segment F does not overlap with the set of used words, and is therefore unaffected. The set of used words now spans words 10 through 69.
4. Segment A completely falls within the set of used words. Therefore, the resulting segment A' is empty, and is thus discarded.
5. Segment E does not overlap with the set of used words, and thus is unaffected. The set of used words now spans words 10 through 69, plus words 80 through 94
6. Segment D overlaps with the used words set on words 90 to 94. Hence, the resulting segment D' only includes words 95 to 99.

At the end of the procedure, the final segments are:

1. Segment B': From word 10 to 29.

2. Segment C': From word 30 to 44.
3. Segment D': From word 95 to 99.
4. Segment E': From word 80 to 94.
5. Segment F': From word 45 to 69.

2 New Scoring Metric

In this section, we propose a new scoring metric for prediction segments, based on overlap quality and effectiveness predictions, which builds on the original metric for the Feedback Prize 1 competition. To this end, we first re-introduce the original metric for greater clarity.

2.1 Original Metric

In Feedback Prize 1.0, the original metric consists of a threshold-based F1 measure computed for each of the 7 discourse element types[2]. In particular, for each of the 7 discourse element types, the metric computes an F1 score as follows:

1. Consider all predicted segments for the given discourse element, and compare pair-wise with each ground truth segment.
2. If a predicted segment overlaps with 50% or more of its words with the ground truth, and covers 50% or more of the words in the ground truth segment, the prediction is counted as a true positive (TP).
3. Were a ground truth segment to satisfy the earlier criteria with more than one prediction segment, then only the segment with the "most" overlap (defined as the maximum of both prediction and ground truth overlap proportions) is accepted, and all other matching segments are considered to be false positives (FP).
4. All prediction segments not satisfying the overlap criteria with any ground truth segment are also considered false positives.
5. All ground truth segments which have not been matched by a satisfactory prediction segment are considered false negatives (FN).

Finally, all seven class-wise F1 scores are averaged to return an overall macro-F1 score for the segmentation task.

Limitations. This metric matches prediction segments to ground truth segments with a reasonable heuristic. However, the matching does not evaluate predictions' quality, e.g., more overlap is better, instead only providing a 0/1 decision

[2]This could be one of lead, concluding statement, position, claim, counterclaim, evidence, and rebuttal.

for each prediction. Moreover, this metric does not include other criteria pursued in later Feedback Prize competitions, such as a 3-class discourse element effectiveness label, and does not penalize biased model performance with respect to student demographics. Finally, the metric is agnostic to runtime, and thus would not differentiate faster, well-performing models from slower ones.

2.2 Incorporating Effectiveness

To incorporate effectiveness into the original FP1 metric, we primarily change the operation of Step 2 above. In particular, a segment satisfying the (unchanged) overlap criteria is no longer automatically classified as a true positive. Instead, we define a *fractional* true positive score, in which the fraction derives from the quality of the effectiveness prediction.

More concretely, consider a hypothetical prediction segment P and a matching ground-truth segment G. In the original FP1 metric, since P is matched to G, we consider P as a true positive[3]. In this extension, we propose a segmentation weight $w \in [0, 1]$, which is the fraction of a TP given simply for matching the ground-truth segment G. For the remaining $1 - w$, we use an effectiveness-based criterion. To this end, we propose to use the predicted probability assigned at P for the correct effectiveness label. For example, should G be considered an effective segment, and P predict "effective" with probability 0.8, then P is assigned a further TP fraction of $(1 - w) \times 0.8$, taking its overall TP fraction to $w + (1 - w) \times 0.8$. Finally, the fractional loss of TP due to effectiveness is added to the False Negative (FN) score, as if part of the ground truth has been missed.

Formally, given a matching prediction segment P with a 2-dimensional effectiveness label distribution $D = (d[0], d[1])$ and a ground-truth segment G, annotated with an effectiveness label $E \in \{0, 1\}$, the true positive fraction assigned to P, denoted TP_P is:

$$\mathrm{TP}_P = w + (1 - w) \times d[E].$$

Moreover, the false negative proportion FN_P is:

$$\mathrm{FN}_P = (1 - w) \times (1 - d[E]).$$

Please note that the remainder of the metric remains unchanged. Unmatched predictions (either non-matching segments or segments that are not the best match for a given ground-truth segment) continue to be treated as false positives. Furthermore, all unmatched ground-truth segments are considered to be false negatives.

As this extension reduces the number of true positives returned by the metric, the resulting score is less than or equal to the original metric score. Moreover, this metric reduces exactly to the original metric should all effectiveness predictions be perfect. All in all, this metric reduces TP by a penalty, which itself is exactly added back into FN, all while keeping FP unchanged.

[3]For simplicity, we assume P is the only match to G, to avoid Step 3.

2.3 Accounting for Overlaps

To account for overlap quality in the shovel-ready competition metric, we propose an analogous solution (fractional TP) to how we incorporate effectiveness scores. Concretely, we can adapt the TP equation to also include a quality criterion on the w coefficient. In particular, we can use the exact same criterion for ranking matching predictions proposed in Step 3 of the original metric, namely the maximum overlap proportion.

Formally, let $0.5 \leq o_P \leq 1$ be the overlap proportion of P's words with respect to the ground-truth G, and let o_G be the analogous proportion of G with respect to P. In Step 3 of the original metric, predictions matching with G are ranked based on $\max(o_P, o_G)$. Hence, a simple extension to include overlap quality is the following:

$$\text{TP}_P = w \times (\max(o_P, o_G)) + (1 - w) \times d[E],$$

with FN_P defined analogously as $1 - \text{TP}_P$.

Note however, that the overlap quality metric used by the original FP1 competition is problematic, as the maximum overlap elevates several otherwise poor predictions. For example, a short prediction P with only 50% of the words of G, and which is fully contained inside of G, would obtain a score of 1, despite missing all the rest of G. Hence, we adopt the **intersection-over-union (IoU)** score between P and G as the quality metric going forward in the shovel-ready competition.

IoU counts the number of words appearing in both, i.e, the intersection, then counts the total number of words appearing in either P or G, i.e., the union, and returns the ratio between the two as an overlap measure. Notice how the former example would be addressed by IoU, as that prediction would only obtain a score of 0.5. IoU naturally fits into the earlier metric proposal. In particular, we can define the fractional TP score as:

$$\text{TP}_P = w \times (\text{IoU}(P, G)) + (1 - w) \times d[E].$$

2.4 Changing the Overlap Threshold

In the original metric, predicted segments are deemed a true positive based on a threshold of 50% (see Step 2 for details). However, this overlap threshold enables a corner case in which scoring ambiguity arises. To illustrate, consider the example in Figure 1.

In this example, we have two contiguous 4-word prediction segments, and another 2 contiguous 4-word ground truth segments. The ground truth is in fact shifted by 2 words to the right relative to the predicted segments, resulting in the fact that the rightmost prediction segment satisfies the Step 2 threshold for both ground truths. Hence, it is ambiguous how best to match segments to ground truths. In particular, it is unclear how to select the correct segment with a concrete rule: If the left prediction is matched to the leftmost ground truth, then a perfect matching is obtained. However, matching the leftmost

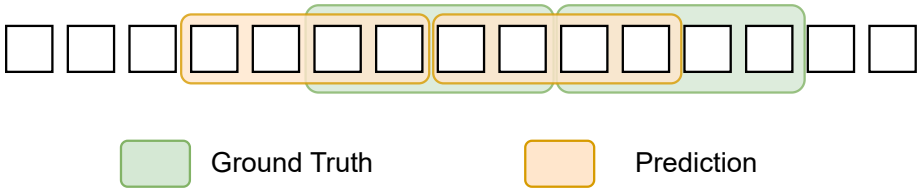

Figure 1: Corner case for the 50% threshold. In this case, the rightmost prediction matches with both ground truth segments, and the choice of matching clearly affects the overall score.

ground truth to the equally matching rightmost prediction results in the loss of a match.

Overall, Step 3 leaves a corner case when the overlap metrics result in an identical score, which in turn leads to a non-deterministic decision that affects the final scores. This problem is in fact exacerbated by overlaps in prediction segments (see Section 1), as this increases the likelihood of multiple matches to a ground truth segment. Hence, we need a clear and unambiguous means to decide on matching, or, failing that, to avoid this scenario altogether. Going forward, we opt for the latter as this greatly simplifies the metric implementation.

To eliminate the possibility of multiple matches, we first remove prediction overlaps in pre-processing. This implies that a ground truth segment can now at most match with 2 different predictions at a ratio of 50% each[4]. To bring this maximum number of matches down to 1, we therefore simply need to increase the threshold to any number strictly above 50%. As a result, we opt to raise the threshold to **51%**[5].

2.5 Scoring Examples.

In this section, we provide a series of examples to illustrate this overall metric. In these examples, we set the fractional TP weight $w = 0.5$, and thus give equal weight to segment matching and to effectiveness prediction on matched segments. We also use an overlap threshold of 51% as described in the previous section. We now consider a set of five working examples, illustrated in Figure 2. For completeness, we provide scores for both overlap quality scores (max overlap and IoU) to highlight the difference.

Example (a). In this example, the prediction segment P overlaps with the ground truth G on 9 out of the 10 words. More precisely, $o_P = 0.9$ and $o_G = 0.9$. Thus, P matches with G and we can compute a fractional TP score. In terms of effectiveness, P produces a score of 80% for the correct label (Effective), and

[4]Note that overlaps should also not exist in the ground truth segments for this to hold

[5]We have validated this choice empirically on a benchmark model and noted this results in a minimal loss of performance of less than 0.01, which confirms the existence of this problem in practice, but also reassuringly highlights its relative rarity.

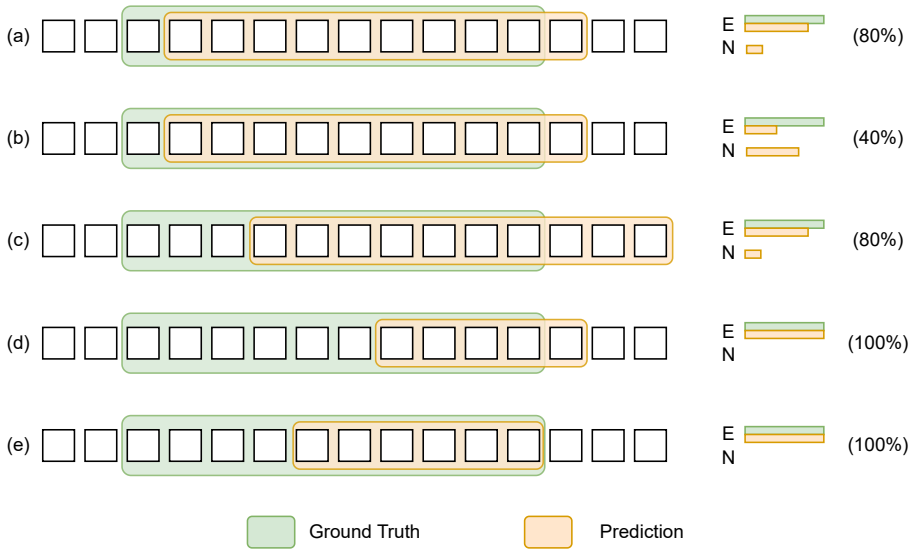

Figure 2: Examples for the overall metric computation. For each example, ground truth segments (green) and predictions (yellow) are shown on the left over a set of words (white boxes). On the right, the effectiveness label (green) and predictions (yellow) for the classes Effective (E) and Non-Effective (N) are shown in a bar chart, along with the predicted probability of the correct class between brackets.

thus the TP fraction assigned to P is:

$$\mathrm{TP}_P = 0.5 \times \max(0.9, 0.9) + (0.5) \times 0.8 = 0.85.$$

If we instead use **IoU** as the overlap measure, the max operation is replaced by $\frac{9}{11}$, as P and G intersect on 9 words and jointly cover 11 words. This leads to the following fractional TP:

$$\mathrm{TP}_P = 0.5 \times \frac{9}{11} + (0.5) \times 0.8 \simeq 0.809$$

.

Example (b). This example uses identical segments as Example (a), but has a worse effectiveness prediction assigned to P. As P is further away from the true effectiveness class (it only assigns 40% likelihood to the "Effective" label), it will receive a lower fractional TP score. Concretely, using max overlap:

$$\mathrm{TP}_P = 0.5 \times \max(0.9, 0.9) + (0.5) \times 0.4 = 0.65.$$

Analogously, for IoU:

$$\mathrm{TP}_P = 0.5 \times \frac{9}{11} + (0.5) \times 0.4 \simeq 0.609.$$

Example (c). In this example, we maintain the effectiveness predictions from Example (a), but have a worse overlap quality. Indeed, segments P and Q only overlap on 7 words out of 10, thus $o_P = 0.7$ and $o_G = 0.7$. This leads to the following fractional TP score using max overlap:

$$\text{TP}_P = 0.5 \times \max(0.7, 0.7) + (0.5) \times 0.8 = 0.75.$$

Moreover, in this example, IoU is now $\frac{7}{13}$, leading to the corresponding score:

$$\text{TP}_P = 0.5 \times \frac{7}{13} + (0.5) \times 0.8 \simeq 0.669.$$

Example (d). In this example, P perfectly matches the effectiveness label of G, but does not sufficiently overlap with it. Indeed, $o_P = 0.8$ and $o_G = 0.4$, and therefore P does not match to G as $o_G < 0.5$. As a result, $\text{TP}_P = 0$ using both max overlap and IoU, as the segment is not accepted as a match to G. Note that effectiveness comparison *is not well-defined* in this scenario, as the two segments do not share a meaningful number of words within the essay for effectiveness comparisons to be sound[6].

Example (e). In this example, P also perfectly matches the effectiveness label of G. However, its overlap is now sufficient: $o_P = 1$ and $o_G = 0.6$, satisfying the matching criterion. Therefore, P is assigned a fractional TP score. Using max overlap, this score is:

$$\text{TP}_P = 0.5 \times \max(1, 0.6) + 0.5 \times 1 = 1.$$

This clearly is not a good score to obtain, as P misses 40% of the words in G! Moreover, P is given a higher score than in Example (a), which is arguably a better overall prediction. By contrast, using IoU as the overlap measure, we obtain an IoU score of 0.6, which leads to a more reasonable fractional TP. Indeed,

$$\text{TP}_P = 0.5 \times 0.6 + 0.5 \times 1 = 0.8.$$

For convenience, we summarize all example scores in Table 1.

3 Accounting for Bias

In the Feedback Prize competitions, several winning models exhibit substantial biases across different student demographics. Unfortunately, such biases were not taken into account when computing scores for the winning models.

Concretely, consider a metric τ applied to a model's predictions[7]. This metric normally applies over the entire test set, with average performance reported

[6] As an extreme point of this argument, consider two fully disjoint segments. Clearly, the effectiveness prediction of one segment does not relate to the other, as they do not even include the same text!

[7] Without loss of generality, we assume that a higher score from τ corresponds to improved performance.

Example	max	IoU
a	0.85	0.809
b	0.65	0.609
c	0.75	0.669
d	0	0
e	1	0.8

Table 1: Summary of fractional TP scores for prediction segment P using max and IoU overlap scores on the examples of Figure 2.

across all test data points. However, this setup does not consider demographic sub-sets of the test set, over which we ideally would like to observe identical performance. In particular, given an overall test population $\mathcal{P}$ that can be partitioned into, e.g., two equally sized subset populations $\mathcal{P}_1$ and $\mathcal{P}_2$ ($\mathcal{P}_1 \cup \mathcal{P}_2 = \mathcal{P}$), a model m where $\tau(m, \mathcal{P}_1) = 0.7$ and $\tau(m, \mathcal{P}_2) = 0.7$ will be ranked identically to another model n, for which $\tau(n, \mathcal{P}_1) = 0.6$ and $\tau(n, \mathcal{P}_2) = 0.8$ respectively, despite m being more consistent than n.

To account for bias against certain sensitive subsets, we propose to apply the metric τ over *sub-populations*, but then to aggregate scores in a *weighted* fashion that penalizes variability among sub-population scores, i.e., bias. The motivation for this is two-fold:

1. Given a (theoretical) model with zero bias, weighted and uniform averages yield identical results. Therefore, less biased models will be less severely penalized.

2. A model with high bias will have sub-populations with better than average performance, as well as other sub-populations where it is lower than average. By weighting bad subsets more, we directly penalize the model for its inconsistent, biased performance.

To illustrate the idea of weighted averaging, we consider the opposite extreme to uniform averaging: selecting the **minimum population score** as the overall score. Going back to our original example with subsets $\mathcal{P}_1$ and $\mathcal{P}_2$, computing the overall score of model m as $\min(\tau(m, \mathcal{P}_1), \tau(m, \mathcal{P}_2))$ yields 0.7, identically to the uniform average, but yields 0.6 for model n, directly penalizing it for its skewed performance profile.

In practice, applying the minimum function to aggregate over sub-population scores may be too harsh. Therefore, we propose to use softer minimum functions, e.g., softmin, whose strictness can be tuned with a temperature parameter α.

Formally, a weighted model score $\nu(m, \mathcal{P})$ can be given by:

$$\beta_i = \frac{|\mathcal{P}_i| e^{-\alpha \times s_i}}{\sum_{j=1}^{k} |\mathcal{P}_j| e^{-\alpha \times s_j}}, \text{ and}$$

$$\nu(m, \mathcal{P}) = \sum_{j=1}^{k} s_i \times \beta_i,$$

where $s_i, i \in \{1, \ldots, k\}$ are positive real-valued metric scores over k testing sub-populations, i.e., $s_i = \tau(m, \mathcal{P}_i)$, and $|\mathcal{P}_i|$ denotes the size of the i^{th} sub-population.

Notice that the above weighted softmin function elegantly captures the two aforementioned extreme cases. Indeed, when the temperature is set to 0, softmin reduces to uniform averaging. Conversely, when temperatures goes to plus infinity, the softmin function asymptotically approaches a hard minimum.

4 Incorporating Efficiency

We propose to account for model efficiency using a *relative reward-based system* to incentivize direct efficiency competition between submissions. In particular, we propose the following relative reward scheme:

1. Run all submissions on the test set and record all of their runtimes (on identical hardware) and their original metric scores.

2. Sort all entries by their performance and by their runtime, and award the fastest submission among eligible submissions a 5% score increase. In this setting, eligibility is tied to *performance*. Indeed, we deem all models within 5% of the best performance score[8] to be eligible models for the efficiency boost.

3. For eligible submissions that are at most 20% slower than the fastest submission, award a score increase linearly proportional to the slowdown percentage. More formally, given a fastest runtime s and a slower model runtime $t > s$, if $\frac{t}{s} \leq 1.2$, award a score increase of $\big(5 - 25 \times (\frac{t}{s} - 1)\big)\%$. For instance, a top-$k$ model that is 10% slower is awarded a 2.5% increase, and a model 15% slower is awarded a 1.25% increase.

This reward system brings several advantages. First, it does not set an absolute efficiency target, but instead drives all submissions to compete to make faster submissions to unlock a scoring boost. Second, the boost (5%) is generous and can substantially change the ranking of a submission. For context, over 30 of the top entries in the Feedback Prize 1 were within 5% of the best score, and thus

[8] Formally, given a gold-standard model M with top performance α and a worse-performing model N with a score β, N is eligible for the boost if and only if $\frac{\alpha}{\beta} \leq 1.05$. In theory, this enables N to overtake M for the best score if M takes 20% or more time to run relative to N.

this mechanism would give slightly less accurate but much faster models a much higher chance of winning. Third, the eligibility mechanism restricting the boost to sufficiently good models, i.e., only models within 5% of the best performance, offers an important safeguard preventing exploitation and trivialization of the reward, e.g., by submitting a null code or a print statement.

4.1 Examples

In this section, we consider two examples to illustrate the function of the efficiency reward mechanism.

Example (a). Consider the leaderboard shown in Table 2. In this leaderboard, the Red model is the best-performing, and runs in 50 seconds. All other models can surpass Red with a 5% boost, and thus they are all eligible for the reward. Among all models, Yellow is the most efficient, and thus is given the 5% boost, raising its score to $0.58 \times 1.05 = 0.609$. Red is 25% slower than Yellow, and Green is 20% slower, and thus they receive no boosts. By contrast, Blue is 12.5% slower than Yellow, and thus receives a boost of $5 - 25 \times 0.125 = 1.875\%$, and thus obtains a boosted score of $0.595 \times 1.01875 \simeq 0.606$. As a result, the final ranking is 1) Yellow, 2) Blue, 3) Red, and 4) Green. Hence, the reward system enables Yellow to jump from fourth to first due to its much better efficiency.

Table 2: Efficiency Example (a).

Model	Metric Score	Runtime	Boosted Score
Red	0.6	50	0.6
Blue	0.595	45	0.606
Green	0.59	48	0.59
Yellow	0.58	40	**0.609**

Example (b). We now consider the leaderboard shown in Table 3. In this leaderboard, the Red model is also the best-performing. However, only one model, Green, can surpass Red with a 5% boost, and thus only Red and Green are eligible for the reward. Hence, Blue and Yellow, despite being faster than Green and Red, are not considered in the reward mechanism. Between Red and Green, Green is the fastest, and thus Green receives a 5% boost, taking its score to $0.59 \times 1.05 \simeq 0.62$. However, Red is only 5% slower than Green, and thus also receives a boost of $5 - 25 \times 0.05 = 3.75\%$, leading to a boosted score of $0.6 \times 1.0375 \simeq 0.623$, preserving its top rank. As a result, the final ranking remains 1) Red, 2) Green, 3) Blue, and 4) Yellow.

Table 3: Efficiency Example (b).

Model	Metric Score	Runtime	Boosted Score
Red	0.6	50.4	**0.623**
Green	0.59	48	0.62
Blue	0.55	45	0.55
Yellow	0.25	5	0.25

This example shows two main properties of the reward mechanism. First, this mechanism rewards faster models commensurately with their relative speed. In Example (a), the boost is sufficient to decide a new winner, but this is not the case in Example (b), as the fastest model is not much faster than the better models ahead of it in the standard ranking. Second, the mechanism is only applied to competitive models, which incentivizes competition at the top of the leaderboard and prevents a trivialization of the reward. Indeed, were all models eligible in Example (b), then Yellow, a very poor model that is extremely fast, e.g. a single decision tree, would simply take the 5% boost and effectively eliminate the efficiency incentive for top submissions. In fact, without the filtering mechanism, it would only take one trivial submission, e.g., a print statement, to virtually eliminate any meaningful rewards.

5 Putting It All Together

Given all the components discussed above, we now present the flow through which a holistic metric can be computed for the shovel ready competition:

1. First, all submitted models are run on an identical hardware configuration, e.g., a cloud server with a sufficiently performant GPU. Their submission files are produced and their runtimes are recorded.

2. Second, the submission files for each entry are pre-processed to remove overlaps, in line with Section 1.

3. Third, the submission files are partitioned based on essay writer demographics to compute metric scores for key sub-categories (ELL, economic disadvantage, etc.) using the new scoring metric (cf. Section 2).

4. Fourth, all sub-category scores are aggregated into a holistic evaluation score using the bias mechanism (softmin, cf. Section 3) at an appropriately selected temperature (currently selected to be 50-80). This result in a single score for each submission.

5. Finally, all submission scores and runtimes are passed through the reward mechanism of Section 4 to obtain a final competition leaderboard[9].

As a result, the competition scoring system incorporates bias penalization, efficiency incentives, and a robust metric. The overall computation flow is illustrated in Figure 3.

[9]Note that the leaderboard to be displayed during the competition will only consist of performance-based scoring up to the bias aggregation step, with the efficiency boosts only applied following the conclusion of the competition. Hence, for convenience, we provide a Colab notebook including code to run the efficiency boosting mechanism locally (as well as the current examples computed in the notebook for more clarity) for competitors to estimate their own current and eventual boosts. You can access this here.

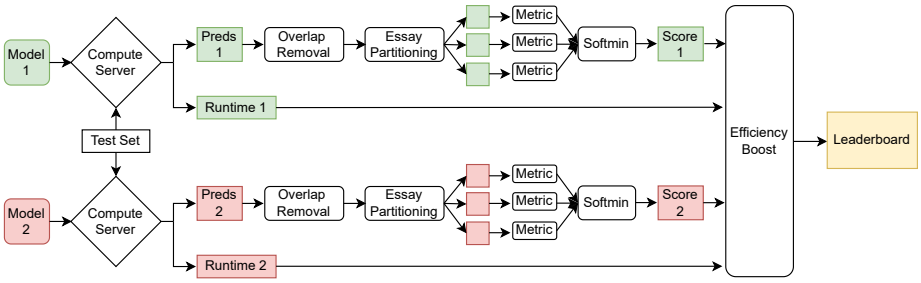

Figure 3: The computational flow of the Shovel Ready competition scoring system, illustrated with only 2 models for simplicity. Both models run on the same test set using an identical compute server to obtain runtimes and predictions for each model. The latter are then processed, partitioned, scored, with scores then re-aggregated into a single score per model using the softmin bias weighting function. Finally, all single model scores and runtimes are fed into the efficiency boost system to obtain a final competition leaderboard.