FRACTAL: Fine-Grained Scoring from Aggregate Text Labels

Yukti Makhija[◇†] **Priyanka Agrawal**[◇] **Rishi Saket**[◇†] **Aravindan Raghuveer**[◇†]
[◇]Google DeepMind
{yuktimakhija, priyankagr, rishisaket, araghuveer}@google.com

Abstract

Fine-Tuning of LLMs using RLHF / RLAIF has been shown as a critical step to improve the performance of LLMs in complex generation tasks. In such methods, typically the responses are sampled from LLMs and human or model feedback is provided at the response level. The feedback is then used to align the LLMs to prefer decoding paths that will agree with the human feedback. Recent works Amplayo et al. [2022], Wu et al. [2023] indicate that sentence-level labels provide more accurate and interpretable feedback for LLM optimization. In this work, we propose FRACTAL a suite of models to disaggregate response-level labels into sentence-level (pseudo-)labels through a Multiple Instance Learning (MIL) formulation, novel usage of prior information and maximum likelihood calibration. We perform close to 2000 experiments across 6 datasets and 4 tasks that show that FRACTAL can reach up to 93% of the performance of the fully supervised baseline while requiring only around 10% of the gold labels. Furthermore, in a downstream eval, employing these sentence-level pseudo scores in RLHF on the Question Answering task leads to 6% improved performance. Our work is the first to develop response-level feedback to sentence-level scoring techniques, leveraging sentence-level prior information, along with comprehensive evaluations on multiple tasks as well as end-to-end finetuning evaluation.

1 Introduction

Large language models (LLMs) are being increasingly used for various generation tasks like generate text Gero et al. [2022], seek facts, answer complex queries Adiwardana et al. [2020], Menick et al. [2022], and perform logical reasoning tasks Kojima et al. [2022]. The improvements to LLMs rely heavily on their evaluation and preference feedback, typically from humans or automated model-based scoring Ouyang et al. [2022], Touvron et al. [2023]. However, such feedback has typically been taken at the response level, enabling efficient and cost-effective assessments of overall output quality.

An emerging body of research Amplayo et al. [2022], Lightman et al. [2023] suggests that the sentence or step-level evaluation is more reliable over response-level evaluation. Finer-grained feedback precisely localizes the strengths and weaknesses within a generated response. It further provides greater interpretability, allowing for more targeted LLM fine-tuning by highlighting the specific portions of a response that contribute to its overall quality. Wu et al. [2023] has shown that collecting finer-grained human feedback results in considerably improved RLHF.

However, collecting fine-grained annotations adds significant cost due to the added quantity and precision of human labor needed. Even in situations where it is feasible to directly collect fine-grained feedback, doing so for the Side-by-Side (SxS) feedback Ouyang et al. [2022] remains challenging.

[†]work done while at Google Research

38th Workshop on Fine-Tuning in Machine Learning (NeurIPS 2024).

In this paper, we argue that it is possible to convert coarse response / paragraph level labels provided by humans into fine-grained sentence level labels. We propose FRACTAL, a suite of modeling based techniques to dis-aggregate response-level labels into sentence-level *pseudo*-labels that accurately reflect the underlying quality distribution within a larger response.

We show across 6 datasets and 4 tasks that FRACTAL can reach upto 93% of performance of a sentence-level model that uses 10X the number of labels as the FRACTAL model. Tab. 1 shows two examples of how FRACTAL converts a gold response label into precise sentence level labels. We also apply FRACTAL to a the fine-grained RLHF setting and show FRACTAL is able to better glean information from the preferences and thereby provide a 6.2% boost to performance over vanilla preference RLHF. To the best of our knowledge, FRACTAL is the first approach to comprehensively study the task of fine-grained scoring from aggregate text and demonstrate practical applicability on Fine-Grained RLHF.

As in supervised training, the first component of FRACTAL is a methodology to train a model on the response-labels to predict the scores (label probabilities) for sentences. For this we leverage and build upon techniques from multiple instance learning (MIL) and learning from label proportions (LLP) (see Sec. A for previous work on MIL and LLP). These have been used to train predictive models on datasets partitioned into *bags* or sets of *instances*. For the text-generation tasks, we model each response as a bag and its instances are the constituent sentences of the response. Each bag has an aggregated label i.e., bag-label which is assumed to be derived from the (unknown) instance-labels of the bag via an aggregation function. The instance-labels correspond to the sentence-labels and the aggregated bag-label is the aggregated response-label. In MIL, the aggregation is the MAX or MIN of binary or ordinal instance-labels – applicable to question-answering (relevance), summarization and math reasoning tasks – while LLP, which models the bag-label as the AVG of the instance-labels, is applicable to retrieval tasks. To estimate the instance-labels from bag-labels, a standard technique in MIL and LLP is *bag-loss* which minimizes a loss between the bag-labels and the aggregated instance-predictions, summed over all bags. While bag-loss is usually a strong baseline, the use of only bag-labels is seen to be insufficient supervision. To this extent, we propose use of *prior* distribution on the instance-label modeled from application-specific information.

Task	**Retrieval | MultiSpanQA Dataset**
Input Query	Who were the last people to be hanged in england?
Retrieved Context (Response)	Nevertheless , there remained a working gallows at HMP Wandsworth , London , until 1994 , which was tested every six months until 1992 . This gallows is now housed in the Galleries of Justice in Nottingham, England. In the United Kingdom : on 13 August 1964 , Peter Anthony Allen , at Walton Prison in Liverpool , and Gwynne Owen Evans , at Strangeways Prison in Manchester , were executed for the murder of John Alan West on 7 April that year . Scotland : Henry John Burnett , 21 , on 15 August 1963 in Craiginches Prison , Aberdeen , for the murder of seaman Thomas Guyan.
Gold Response Label	**1** i.e. Retrieved Context is relevant to the Input
Sentence Labels - FRACTAL	**[0, 0, 1, 0]** i.e. third sentence contains the answer
Task	**Question Answering | QA-Feedback Dataset**
Input Query	What is the name of the pirate in spongebob?
Input Context	SpongeBob SquarePants - Puff endures one of SpongeBob's crashes or is otherwise frightened, she puffs up into a ball. Special episodes of the show are hosted by a live action pirate named Patchy and his pet parrot Potty, ...
Response 1	The SpongeBob SquarePants is an American animated television series based on the cartoon series SpongeBob SquarePants. A live action pirate named Patchy and his pet parrot Potty, who host special episodes of the show, are presented in a dual narrative with the animated stories. Burger Beard is the name of the pirate in the SpongeBob SquarePants movie.
Response 2	The SpongeBob Movie: Sponge Out of Water is a 2015 American live-action/animated absurdist comedy film based on the animated television series SpongeBob SquarePants. It follows a pirate named Burger Beard, who steals the Krabby Patty secret formula using a magical book that makes any text written upon it come true. SpongeBob and his friends must travel to the surface to confront Burger Beard and get the formula back.
Gold Preference Label	**1** i.e. Response 1 is better than Response 2
Sentence Labels - FRACTAL	Response 1: **[1, 1]** ; Response 2: **[1, 0, 0]** i.e. Sentences 2 and 3 of Response 2 are inaccurate.

Table 1: Our method, FRACTAL, dis-aggregates response-level gold label into fine-grained sentence-level labels. We can then use these sentence-level labels for finetuning LLMs. These examples are for the tasks of Retrieval and Question Answering from MultiSpanQA and QA-Feedback datasets.

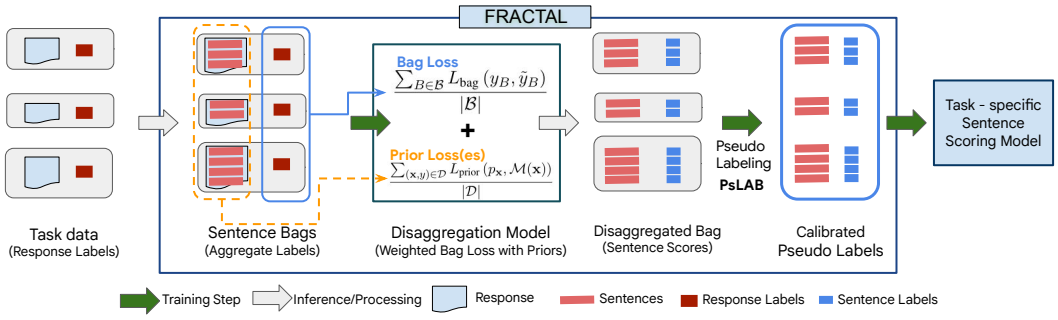

Figure 1: Overview of our proposed method, FRACTAL. Input is a set of responses each with a response label. A response is a bag of sentences. The output is a model that can predict the score for each sentence in a response. The semantic meaning of a score depends on how the response label was defined. FRACTAL consists of two key components a) Bag-level loss functions and model training (Section 3.1), and b) Max-Likelihood Pseudolabeling (Section 3.2)

We make the following contributions:

1. We propose a novel framework, FRACTAL, to disaggregate response labels into constituent sentence-labels. We formulate the fine-grained prediction as a Multiple Instance Learning (MIL) and Learning from Label Proportions (LLP) task. This abstraction allows us to leverage baselines proposed for MIL/ LLP . Through our experiments, we show that this formulation alone is not sufficient to demonstrate strong performance.
2. We introduce enhancements by adding priors over the instances. We propose to add two types of priors for every instance in a bag based on document-sentence similarity scores and correlations between sentences. The baseline MIL / LLP methods are augmented with prior information as new loss terms as show in Sec. 3.1. To the best of our knowledge, use of such priors to improve performance of MIL/LLP methods has not been studied before.
3. We develop pseudo-labeling strategies to calibrate instance-level model predictions into labels which are consistent with the response-level labels, allowing us to train the model on the derived pseudo-labels (refer Sec. 3.2). Our ablation experiments demonstrate that both the prior inclusion and pseudo-labeling steps significantly help improve performance.
4. In order to study the performance of FRACTAL, we formulate a wide variety of tasks shown in Sec. 4: retrieval, question answering, summarization, and math reasoning across 6 datasets. We define the required formulations for disaggregating response-level labels to sentence-level labels applicable to these tasks. In intrinsic evals, FRACTAL achieves up to 93% of the performance of the fully supervised baseline while requiring only around 10% of the number of labels.
5. In an extrinsic finetuning eval, FRACTAL also improves the performance of preference RLHF by 6% through fine grained labeling.

2 Preliminaries

As mentioned in Section 1, we consider text-generation tasks where each response can be modeled as a bag whose instances are the constituent sentences of the response. The bag-label is the response-label while the sentence-labels correspond to the respective instance-labels. In the following, we formally define the notion of instances, bags and their labels.

Let $\boldsymbol{\mathcal{X}}$ be the underlying set of instances and $\mathcal{Y}$ be the label-set which is $\{0, 1, \ldots, C\}$ for some $C \in \mathbb{Z}^+$ where $C = 1$ for binary and $C > 1$ for integer labels respectively. A dataset is a collection of labeled instances.

A bag B is a subset of $\boldsymbol{\mathcal{X}}$ and y_B denotes its label which is thought to depend on the labels of the instances in B via an *aggregation function* AGG which maps tuples with elements from $\mathcal{Y}$ to $[0, C]$. Specifically, if $B = \{\mathbf{x}_1, \ldots, \mathbf{x}_k\}$ and y_i is the label of $\mathbf{x}_i$ ($i \in [k]$), then $y_B = \mathsf{AGG}(\mathbf{y}_1, \ldots, \mathbf{y}_k)$. Typically, AGG is either MIN, MAX or AVG.

We consider *prior information* about the labels on individual instances, for e.g. through unsupervised modeling. For each $\mathbf{x}$ in the dataset, its *point*-prior $p_{\mathbf{x}} \in [0,1]$ is prior for $y_{\mathbf{x}}/C$ where $y_{\mathbf{x}}$ is the underlying label of $\mathbf{x}$ and $\{0,\ldots,C\}$ is the label set as defined above. The *pair*-prior for a pair $(\mathbf{x},\mathbf{z})$ of instances is given by $p_{\mathbf{xz}} \in [0,1]$ and measures the correlation between $\mathbf{x}$ and $\mathbf{z}$. Section 4 and Table 3 provide the specifics of the prior information for various datasets and tasks in our experiments.

Some applications provide *preference* bag-labels which encode comparisons between pairs of bags. Specifically, for a pair of bags (B_1, B_2) the preference bag-label $y_{B_2>B_1}$ is 0 if $y_{B_1} > y_{B_2}$, and 1 if $y_{B_1} < y_{B_2}$. There are no bag-labels, only preference bag-labels for some pairs of bags.

Modeling Task. Given as input a collection $\mathcal{B}$ of pairwise-disjoint (i.e., non-overlapping) bags along with their bag-labels (or preference bag-labels), possibly along with the priors $\{p_{\mathbf{x}}\}$ or $\{p_{\mathbf{xz}}\}$, the goal is to output a model predicting a score for each instance in $\boldsymbol{\mathcal{X}}$. In the preference evaluation, we evaluate the model in terms of the accuracy of the preference labels assigned by the model on a test set of bags. In the description of our techniques we will use CE to denote the cross entropy loss.

3 Our Techniques

We present the components of the FRACTAL method along with the BagLoss baseline approach. (Refer Figure 1). The two main components of FRACTAL are (i) model training using bag-level loss functions which incorporate the priors as defined in Section 2, and a (ii) pseudo-labeling technique to use the model predictions to provide instance-level pseudo-labels using which the final model training is trained.

3.1 Training with bag-loss and priors

We train a model $\mathcal{M}$ on the collection $\{B \in \mathcal{B}\}$ of training bags B with aggregate labels y_B. The prediction $\mathcal{M}(\mathbf{x})$ of the model on any instance $\mathbf{x}$ is a probability distribution over the label-set $\{0,\ldots,C\}$, and we denote the probability of label ℓ by $\mathcal{M}(\mathbf{x})[\ell]$. Also, by $\tilde{y}(\mathbf{x})$ we denote the soft-label $\sum_{\ell\in\{0,\ldots,C\}} \ell \cdot \mathcal{M}(\mathbf{x})[\ell]$, assigned by the model to $\mathbf{x}$. Let $\mathsf{probAGG}$ be an extension of AGG to sequences of model-predictions. In particular, $\mathsf{probAGG}$ maps sequences $(\mathcal{M}(\mathbf{x}_1),\ldots,\mathcal{M}(\mathbf{x}_k))$ to a probability distribution over $\{0,\ldots,C\}$, for $k \in \mathbb{Z}^+$. We explicitly define $\mathsf{probAGG}$ for the tasks used in our work in Appendix B.

The baseline BagLoss method optimizes the following loss, for a model $\mathcal{M}$:

$$L_{\text{totbag}}(\mathcal{B},\mathcal{M}) := |\mathcal{B}|^{-1} \sum_{B\in\mathcal{B}} \mathsf{CE}\left(y_B, \mathcal{M}(B)\right) \tag{1}$$

where $\mathcal{M}(B) = \mathsf{probAGG}\left((\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in B}\right)$ is the aggregate prediction of B, and $\mathcal{M}(B)[\ell]$ is probability of label ℓ.

PriorsBagLoss. In our bag loss with priors method, we have an additional loss which incorporates the priors. For point-priors $\{p_{\mathbf{x}}\}$ we have:

$$L_{\text{totpntprior}}(\boldsymbol{\mathcal{X}},\mathcal{M}) := |\boldsymbol{\mathcal{X}}|^{-1} \sum_{\mathbf{x}\in\boldsymbol{\mathcal{X}}} \mathsf{CE}\left(p_{\mathbf{x}}, \tilde{y}(\mathbf{x})/C\right) \tag{2}$$

while for pair-priors $\{p_{\mathbf{xz}}\}$ the loss is

$$L_{\text{totprprior}}(\boldsymbol{\mathcal{X}},\mathcal{M}) := |\boldsymbol{\mathcal{X}}|^{-2} \sum_{(\mathbf{x},\mathbf{z})\in\boldsymbol{\mathcal{X}}^2} L_{\text{prprior}}\left(p_{\mathbf{xz}}, \tilde{y}(\mathbf{x}), \tilde{y}(\mathbf{z})\right) \tag{3}$$

where L_{prprior} is a specialized loss described in Appendix C. The total loss in **PriorsBagLoss** is a convex combination of bag and prior losses:

$$L_{\text{totPB}} = \lambda L_{\text{totbag}} + \lambda_1 L_{\text{totpntprior}} + \lambda_2 L_{\text{totprprior}} \tag{4}$$

for some $\lambda, \lambda_1, \lambda_2 \in [0,1]$ s.t. $\lambda + \lambda_1 + \lambda_2 = 1$. The values λ, λ_1 and λ_2 are selected through a hyperparameter search which is also described in Section 5.

Minibatch based model training. For a given batch size q, learning rate and optimizer as well as hyperparameter $\lambda \in [0,1]$, we train the predictor model $\mathcal{M}$ by doing the following for N epochs and K steps per epoch:

1. Sample a minibatch S of q bags $\mathcal{B}_S \subseteq \mathcal{B}$.
2. Using current model predictions, compute L_{totbag}, $L_{\text{totpntprior}}$ and $L_{\text{totprprior}}$ restricted only to the bags and instances in S, and compute L_{tot}.
3. Using the required gradients from (1), (2) and (3) along with the optimizer and learning rate, update the weights of the model $\mathcal{M}$.

Preference based bag-loss with priors. The approach is similar to that in the previous subsection, where instead of L_{totbag} we have a preference based loss for the pairs of bags S for which preference labels are available. Define $\tilde{y}(B) := \sum_{\ell \in \{0,\ldots,C\}} \ell \cdot \mathcal{M}(B)[\ell]$ be the real-valued soft-label for a bag B. For a pair of bags (B_1, B_2) with $y_{B_2 > B_1}$ be the preference-label, we incorporate the Bradley-Terry model Bradley and Terry [1952] used in previous work , to measure the inconsistency of the predictions with the preference label. Specifically, we define the loss:

$$L_{\text{pref}}(B_1, B_2, y_{B_2>B_1}) := \mathsf{CE}\left(y_{B_2>B_1}, \frac{\tilde{y}(B_2)}{\tilde{y}(B_1) + \tilde{y}(B_2)}\right) \tag{5}$$

$L_{\text{pref}}(B_1, B_2, y_{B_2>B_1})$ is averaged over all pairs in S to obtain L_{totpref} which we refer to as **PrefBagLoss**. The minibatch training now samples pairs of bags and computes L_{totpref} restricted to the sampled pairs. In the priors based augmentation, **PriorsPrefBagLoss**, $L_{\text{totpntprior}}$ and $L_{\text{totprprior}}$ losses remain the same, over all the instances in the minibatch. In this case, the total loss is

$$L_{\text{totPPB}} = \lambda L_{\text{totpref}} + \lambda_1 L_{\text{totpntprior}} + \lambda_2 L_{\text{totprprior}}. \tag{6}$$

3.2 PsLab: Pseudo-labeling

Our pseudo-labeling method, PsLab uses the predictions of the model $\mathcal{M}$ trained as per the techniques described above, to output the max-likelihood instance-level labels for each bag, consistent with the bag-label, i.e., an instance-level pseudo-labeling independently for each bag, which well-defined since the bags are disjoint. We describe PsLab for the binary case of $C = 1$ and only for the MIN aggregation since MAX is equivalent to MIN by flipping the labels.

Case $\{0,1\}$-labels and MIN aggregation. Given bag $B = \{\mathbf{x}_1, \ldots, \mathbf{x}_k\}$ and y_B, we output a pseudolabeling $\Gamma_B : \{\mathbf{x}_1, \ldots, \mathbf{x}_k\} \to \{0,1\}$ as follows: if y_B is 1 then Γ_B assigns 1 to all $\mathbf{x} \in B$.

If not, we first set $\Gamma_B(\mathbf{x}) = 1$ for those $\mathbf{x} \in B$ on which the model prediction $\mathcal{M}(\mathbf{x})$ is biased towards 1. If this results in all Γ_B assigning 1 to all $\mathbf{x} \in B$, then we let $\Gamma_B(\mathbf{z}) = 0$ for some $\mathbf{z}$ on which the model predicts the smallest probability for label 1.

For $C > 1$ PsLab is somewhat more involved and the algorithmic details along with a proof of correctness are included in Appendix M.

Model Training on Pseudo-labels. After computing the pseudo-labels on the training bags, we now have a train-set with labeled instances. The model retrained on this dataset is evaluated for comparative performance.

4 Tasks and Datasets

Task	Dataset	Objective	Train | Test Instances	Train Bags
Long-form QA	QA-Feedback	Preference	93.8k | 9k	14k
Retrieval	FirA	Classification	102k | 20.4k	18k
Retrieval	MultiSpanQA	Classification	47k | 5.7k	5k
Summarization	AquaMuSe	Classification	13.5k | 2.5k	3k
Summarization	WikiCatSum	Classification	209k | 10.7k	45k
Math Reasoning	PRM800K	Classification	663k | 18.5k	98k

Table 2: Summary of the setup used for each dataset

We consider following six datasets covering four tasks listed under Tables 3 and 2. More details on data processing in Appendix D.

Long-form Question Answering. We use QA-Feedback dataset, an SxS preference dataset collected and released by Wu et al. [2023]. These are human preferences on pairs of model generated responses for input questions and relevant passages from ASQA Stelmakh et al. [2023]. The data further contains segment-level annotations, and we make use of "irrelevance, repetition, or incoherence" category for evaluation. The responses are the bags and the preferences are the preference bag-labels.We use AVG as the aggregation function and point-prior is cosine similarity between knowledge passages and each sentence of the response. Cosine-similarity represents the semantic similarity or relevance between a sentence and the context and is an estimate of the sentence

Dataset (Input for training)	Bags	Instances	Task	Prior
QA-Feedback (Question, Knowledge Passage, Pair of Responses, Preference Label)	Both responses are treated as separate bags.	Sentences in a response + question + Knowledge Passages	Learn sentence-level scores for relevance using only the preference bag-label (indicates which response is better). The aggregation function used is AVG	knowledge passage-sentence cosine sim., corr. b/w sentences.
FirA (Paragraph, Query, Relevance Score)	Paragraph	sentence of the paragraph + query	Learn $\{0, 1, 2, 3, 4\}$-valued relevance score for each sentence wrt query. The aggregation function used is MAX	query-sentence cosine similarity
MultiSpanQA (Context, Question, Label (answer present in context)	Context	Sentences of the context + Questions	Identify sentences of the context which contain the answer to the question. We use MAX for this binary classification setup.	query-sentence cosine sim., corr. b/w sentences
AquaMuSe (Documents, Query, Summary, Entailment Label)	Summary	Sentence of the summary + documents + query	Given a query, document and bag-level binary entailment label, determine the non-entailed sentences in a summary. MIN is the aggregate function used in this setup.	doc-sentence cosine sim., corr. b/w sentences
WikiCatSum (Documents, Summary, Entailment Label)	Summary	Sentence of a summary + documents	Given a document and bag-level binary entailment label, determine the non-entailed sentences in a summary. MIN is the aggregate function used in this setup.	doc-sentence cosine sim., corr. b/w sentences
PRM800K (MATH Problem, step-wise solution, Label(correctness)	Solution to the MATH problem	Step of the solution + question	Using the binary aggregate label indicating the correctness of the solution, identify all incorrect steps in the solution.	question-step cosine sim., corr. b/w steps

Table 3: Summary of the bags, labels, instances, annotations and priors for each dataset.

label. Specifically:

$$p_{\mathbf{x}} := \mathsf{cosprior}(\mathbf{x}) = 0.5\left(1 + \langle \mathbf{x}, \mathbf{U}\rangle / (\|\mathbf{x}\|_2 \|\mathbf{U}\|_2)\right) \tag{7}$$

where $\mathbf{x}$ and $\mathbf{U}$ are (embeddings of) a sentence and the relevant passage. Additionally, we incorporate a pair-prior:

$$p_{\mathbf{xz}} := \mathsf{corrprior}(\mathbf{x}, \mathbf{z}) = \left(1 + \rho_{\mathbf{xz}}\right)/2 \tag{8}$$

where $\rho_{\mathbf{xz}}$ is the Pearson's correlation between sentence embeddings $\mathbf{x}$ and $\mathbf{z}$ and represents the probability of two sentences have the same label.

Retrieval. We use two datasets for retrieval tasks: MultiSpanQA and FiRA. MultiSpanQA Li et al. [2022] consists of question and retrieved context pairs, with annotated discontinuous answer spans for the train and validation splits (See Tab. 1 for example). Both the instance and bag labels are $\{0, 1\}$-valued. The FiRA dataset Hofstätter et al. [2020] comprises word-level relevance annotations using $\{0, \ldots, 4\}$-valued labels. We derive the sentence-level scores by taking the word-level average across annotators and then the maximum across all words in a sentence. Similar to the previous setup, we treat the paragraph as a bag, its sentences as instances, and employ MAX as the aggregation function. The instance and bag-level belongs to the set $\{0, 1, 2, 3, 4\}$, with the goal of optimizing a cross entropy loss. For both datasets, we integrate a correlation prior between sentence pairs and a cosine-similarity prior (see (7), (8)) between the query and each sentence of the context.

Summarization. We utilize two datasets: WikiCatSum Perez-Beltrachini et al. [2019] and AquaMuSe Kulkarni et al. [2020]. We adopt the binary entailment metric for this task. The reference summaries already provided in these two datasets serve as the *entailed* summaries[†] with each sentence considered positively entailed. To generate *non-entailed* summaries, we synthesize negatives similar to Yin et al. [2021]. Examples of entailed and non-entailed summaries are provided in Appendix L, along with the method of generation. As in previous tasks, we incorporate sentence-document cosine similarity and sentence correlation priors into our methods. Additionally, we experiment with NLI entailment scores Honovich et al. [2022] as priors for this task.

Math Reasoning. We utilize PRM800K dataset Lightman et al. [2023] releasing step-level annotations for model-generated solutions to MATH problems Hendrycks et al. [2021]. The task at hand is to identify all the incorrect steps in the solution. Similar to previous tasks, we experiment with question-step cosine similarity prior and a correlation prior between steps of the solution.

5 Experiments

We evaluate FRACTAL along with baseline methods on the tasks and datasets described in Sec. 4.

Priors as Baselines. The following methods based on the priors described in Sec. 4 are directly used as baselines to score the sentences as shown in Tab. 4:
Cosine Similarity: In this baseline, the semantic similarity of individual sentences of the response with the input context is used to estimate their relevance score for the task. For this, we compute the cosine similarity (see Eq. (7)) between the corresponding embeddings.

[†]In this work, we do not filter any noise present in the existing data splits.

NLI - Entailment Scorer: For summarization and relevance tasks, we also compute entailment using the NLI scorer from TRUE paper Honovich et al. [2022]. This is a T5x-11B model Raffel et al. [2023] finetuned on several NLI datasets.

Trainable Baselines. These baselines use the bag or instance labels to train models.
BagLoss: This uses BagLoss, L_{totbag}, on bag-labels (or PrefBagLoss $L_{totpref}$ in case of preference bag-labels) described in Sec. 3.
Response-level: For training, this uses entire response as a singleton bag i.e. $|B| = 1$ with x_1 = response. Inference is done on sentences.
Supervised: Trains directly on sentence-labels i.e. Train Instances of Tab. 2 to provide an upper baseline for comparison.

FRACTAL: As described in Sec. 3, this involves PriorsBagLoss (or PriorsPrefBagLoss) based model training using bag-labels (or preference bag-labels) as well priors. In the tables, PriorsBagLoss(λ_1, λ_2) and PriorsPrefBagLoss(λ_1, λ_2) denote the instantiation of these methods with weights λ_1 and λ_2 for the losses corresponding to the point and pair priors respectively (see (7), (8) and Sec. 3.1). For the WikiCatSum dataset, we incorporate NLI entailment scores as a prior, assigning a weight of λ_3 for the corresponding loss term. The weight for bag-loss (or preference bag-loss) is adjusted so that the sum of all the loss weights is 1. PSLAB denotes the performance of the model trained after pseudo-labeling the train-set using the best performing prior augmented bag loss. Note that when we only have preference bag-labels i.e., in the QA Preference Feedback dataset PSLAB is not applicable, and FRACTAL provides the model trained using PriorsPrefBagLoss.

Method	AUC-ROC	AUC-PR	Accuracy
MultiSpanQA (47k | 5.7k)			
Supervised	0.734 ± 0.013	0.358 ± 0.009	0.872 ± 0.042
Cosine Similarity	0.455	0.135	0.851
NLI	0.631	0.366	**0.859**
Response-level Model	0.582 ± 0.113	0.221 ± 0.089	0.851 ± 0.007
BagLoss	0.662 ± 0.069	0.307 ± 0.075	0.849 ± 0.091
FRACTAL*	**0.687 ± 0.062**	**0.329 ± 0.053**	0.843 ± 0.049
FRACTAL/Supervised%	93.59%	91.89%	96.67%
QA Preference Feedback (93.8k | 9k)			
Supervised	0.652 ± 0.008	0.611 ± 0.007	0.691 ± 0.015
Cosine Similarity	0.535	0.526	0.483
Response-level Model	0.491 ± 0.008	0.4643 ± 0.007	0.453 ± 0.015
PrefBagLoss	0.511 ± 0.004	0.53 ± 0.002	**0.524 ± 0.007**
FRACTAL**	**0.537 ± 0.003**	**0.531 ± 0.003**	0.519 ± 0.006
FRACTAL/Supervised%	82.36%	86.91%	75.11%
WikiCatSum (209k | 10.7k)			
Supervised	0.831 ± 0.051	0.889 ± 0.062	0.714 ± 0.048
Cosine Similarity	0.408	0.829	0.362
NLI	0.639	0.817	0.648
BagLoss	0.478 ± 0.065	0.829 ± 0.038	0.569 ± 0.036
FRACTAL*	**0.643 ± 0.031**	**0.875 ± 0.035**	**0.665 ± 0.062**
FRACTAL/Supervised%	77.37%	98.42%	93.14%
AquaMuSe (13.5k | 2.5k)			
Supervised	0.878 ± 0.007	0.925 ± 0.002	0.867 ± 0.008
Cosine Similarity	0.632	0.763	0.649
NLI	0.793	0.889	0.824
Response-level Model	0.695 ± 0.009	0.775 ± 0.007	0.673 ± 0.01
BagLoss	0.747 ± 0.007	0.824 ± 0.005	0.779 ± 0.01
FRACTAL*	**0.815 ± 0.005**	**0.899 ± 0.006**	**0.833 ± 0.01**
FRACTAL/Supervised%	92.71%	97.19%	96.08%
PRM800K (663k | 18.5k)			
Supervised	0.652 ± 0.015	0.935 ± 0.013	0.727 ± 0.022
Cosine Similarity	0.51	0.876	0.516
Response-level Model	0.537 ± 0.057	0.879 ± 0.045	0.535 ± 0.076
BagLoss	0.562 ± 0.024	0.883 ± 0.029	**0.671 ± 0.033**
FRACTAL*	**0.593 ± 0.014**	**0.901 ± 0.006**	0.618 ± 0.016
FRACTAL/Supervised%	90.05%	96.35%	85.01%

Table 4: Evaluations on Test-set (instance-level). Prefix * indicates PSLAB method and ** indicates PriorsPrefBagLoss. Last row for each dataset has % of supervised achieved by FRACTAL. Note: PSLAB is not applicable QA Preference Feedback and FRACTAL is the model trained using PriorsPrefBagLoss.

Model Training Setup. Details about the model architecture for different tasks, training setup and hyperparameter tuning can be found in Appendix F.

5.1 Experimental Results

Tab. 4 and 5 provide the detailed evaluations of the baselines and FRACTAL i.e PSLAB with best performing PriorsBagLoss. For QA Preference Feedback, since PSLAB is not applicable, FRACTAL provides the model trained using PriorsPrefBagLoss. Tab. 6 provides the results for fine-grained RLHF on the QA-Feedback dataset using the framework provided by Wu et al. [2023] by replacing the human annotations (supervised) with relevance label predictions from our FRACTAL model trained only on preference labels. We evaluated the performance of the generated summaries by calculating the ROUGE score between them and the reference summaries across the entire test set. Additionally, we conducted human evaluations for generated outputs of both the finetuned models on 175 samples to determine the precision score. Annotators compare the generated text with the reference output, counting the number of sentences in the generated text that contain information from the reference. Thus, the precision represents the fraction of relevant sentences in the generated output.

Method	MAE	MSE
Supervised	0.283 ± 0.072	0.141 ± 0.088
Response-level Model	0.319 ± 0.047	0.186 ± 0.098
BagLoss	0.304 ± 0.007	0.163 ± 0.002
PriorsBagLoss(0.2, 0.2)	0.294 ± 0.003	0.155 ± 0.001
FRACTAL	**0.293 ± 0.001**	**0.152 ± 0.002**
FRACTAL/Supervised increase%	3.5%	7.8%

Table 5: Test (instance-level) evaluation on FiRA.

Ablations. In Tab. 7 we provide an ablation study on the effectiveness of priors and show how much addition of each prior contributes to the performance of BagLoss shown in Tab. 4. We also adapt our loss functions to ingest instance-level data by adding the instance-level loss term. Tab. 8 shows

the performance of the fully supervised model trained on a randomly sampled subset 20% labeled training instances, along with models trained using our methods on the *hybrid* dataset i.e., the 20% labeled training instances along with the remaining bag-level train-set. Results for these ablations are in Appendix G.

5.2 Discussion

FRACTAL is highly label-efficient while performing close to full supervision. As we can see from Tab. 2, the number of training bags is only a small fraction of (10 to 25 %) of the training instances. Thus, FRACTAL consumes only 10 to 25 % of the labels required to train a fully supervised model. Nevertheless, the classification performance of the FRACTAL method (Tab. 4) shows that it achieves $> 90\%$ of the performance of the fully supervised model for most of the datasets/metrics except for QA Preference Feedback where the it recovers $\approx 80\%$ of the supervised baseline.

Method	ROUGE	Precision
SFT + Preference RLHF	43.759	0.451
SFT + FineGrained RLHF (FRACTAL)	44.087	0.479

Table 6: Fine-grained RLHF on QA-Feedback.

FRACTAL renders more precise sentence-level scores. From Tab. 4, we observe a consistent improvement in the sentence scoring over the BagLoss as well as the Response-level baseline across all these datasets in terms of AUC-ROC, bridging the performance gap between them and the best approach supervised (instance-level trained) model. The use of priors along with pseudo-labeling based model training allows for an improved estimation of task specific score for sentences. It is interesting to note that BagLoss outperforms the Response-level baseline, suggesting that introduction of aggregate loss based methods to estimate sentence-level scores is itself useful. Response-level model trained with large response doesn't generalize well to individual sentences.

Leveraging Prior improves Feedback Disaggregation. Among the key ideas of FRACTAL is to augment BagLoss with cosine-similarity and correlation priors at the sentence-level (see (7) and (8)). As observed from our ablations in Tab. 7 along with the results in Tab. 4, as well as Tab. 5 for FiRA, introducing the prior loss terms, provides substantially improved performance over either using just the BagLoss, or the cosine-similarity/NLI baselines. In effect, our proposed combination performs better than either of its constituents. We hypothesize that the priors provide sentence-level insight which complements the aggregate label based optimization of BagLoss. While performance of FRACTAL is sensitive to the weights of each loss component, during our hyperparameter sweep, we find that the model tends to select an upweighted BagLoss term for most datasets and metrics except for QA Preference Feedback.

Calibrated Pseudo-Labels are more helpful. From Tables 4, 5, 7 we can see that for all tasks applying PSLAB produces outperforms the models trained using PriorsBagLoss. This indicates the effectiveness of calibrating the disaggregated scores such that their aggregate matches the response scores for training the final model.

Downstream Tasks benefit from sentence-level scoring. Wu et al. [2023] showed that using fine grained human labels per segment level across 3 categories irrelevant, untruthful, completeness can help improve performance. In the above setup, we replace the fine grained human labels of relevance with the FRACTAL predictions (Refer Long-Form Question Answering task in Sec. 4). We observe in Tab. 6 that FRACTAL (Row 2) helps improve performance over using only preference labels (Row 1) by around 6.2% in precision and 0.74% in the ROUGE score.

6 Conclusions

Our works casts the problem of deriving sentence-level scores from response-labels for complex text generation tasks as that of learning from aggregated labels in the MIL and LLP frameworks. We propose a novel method FRACTAL, which augments bag-loss using instance level priors to train predictor models, along with a pseudo-labeling technique for improved model training. Extensive evaluations of FRACTAL along with vanilla bag-loss and response-level model training baselines, as well as off-the-shelf scorers demonstrate substantial performance gains from FRACTAL models on six datasets spanning four tasks: retrieval, question answering, summarization, and math reasoning.

References

Daniel Adiwardana, Minh-Thang Luong, David R. So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, and Quoc V. Le. Towards a human-like open-domain chatbot. *CoRR*, abs/2001.09977, 2020. URL https://arxiv.org/abs/2001.09977.

Reinald Kim Amplayo, Peter J. Liu, Yao Zhao, and Shashi Narayan. Smart: Sentences as basic units for text evaluation, 2022.

Stefanos Angelidis and Mirella Lapata. Multiple instance learning networks for fine-grained sentiment analysis. *Transactions of the Association for Computational Linguistics*, 6:17–31, 2018. doi: 10.1162/tacl_a_00002. URL https://aclanthology.org/Q18-1002.

Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernandez Abrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan Botha, James Bradbury, Siddhartha Brahma, Kevin Brooks, Michele Catasta, Yong Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, Clément Crepy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, Mark Díaz, Nan Du, Ethan Dyer, Vlad Feinberg, Fangxiaoyu Feng, Vlad Fienber, Markus Freitag, Xavier Garcia, Sebastian Gehrmann, Lucas Gonzalez, Guy Gur-Ari, Steven Hand, Hadi Hashemi, Le Hou, Joshua Howland, Andrea Hu, Jeffrey Hui, Jeremy Hurwitz, Michael Isard, Abe Ittycheriah, Matthew Jagielski, Wenhao Jia, Kathleen Kenealy, Maxim Krikun, Sneha Kudugunta, Chang Lan, Katherine Lee, Benjamin Lee, Eric Li, Music Li, Wei Li, YaGuang Li, Jian Li, Hyeontaek Lim, Hanzhao Lin, Zhongtao Liu, Frederick Liu, Marcello Maggioni, Aroma Mahendru, Joshua Maynez, Vedant Misra, Maysam Moussalem, Zachary Nado, John Nham, Eric Ni, Andrew Nystrom, Alicia Parrish, Marie Pellat, Martin Polacek, Alex Polozov, Reiner Pope, Siyuan Qiao, Emily Reif, Bryan Richter, Parker Riley, Alex Castro Ros, Aurko Roy, Brennan Saeta, Rajkumar Samuel, Renee Shelby, Ambrose Slone, Daniel Smilkov, David R. So, Daniel Sohn, Simon Tokumine, Dasha Valter, Vijay Vasudevan, Kiran Vodrahalli, Xuezhi Wang, Pidong Wang, Zirui Wang, Tao Wang, John Wieting, Yuhuai Wu, Kelvin Xu, Yunhan Xu, Linting Xue, Pengcheng Yin, Jiahui Yu, Qiao Zhang, Steven Zheng, Ce Zheng, Weikang Zhou, Denny Zhou, Slav Petrov, and Yonghui Wu. Palm 2 technical report, 2023.

Ehsan Mohammady Ardehaly and Aron Culotta. Domain adaptation for learning from label proportions using self-training. In *IJCAI*, pages 3670–3676, 2016.

Ehsan Mohammady Ardehaly and Aron Culotta. Co-training for demographic classification using deep learning from label proportions. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 1017–1024. IEEE, 2017.

Boris Babenko. Multiple instance learning: algorithms and applications. *View Article PubMed/NCBI Google Scholar*, 19, 2008.

Denis Barucic and Jan Kybic. Fast learning from label proportions with small bags. In *Proc. IEEE ICIP*, pages 3156–3160, 2022.

Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952. ISSN 00063444. URL http://www.jstor.org/stable/2334029.

Jatin Chauhan, Xiaoxuan Wang, and Wei Wang. Learning under label proportions for text classification. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12210–12223, Singapore, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.817. URL https://aclanthology.org/2023.findings-emnlp.817.

L. Chen, Z. Huang, and R. Ramakrishnan. Cost-based labeling of groups of mass spectra. In *Proc. ACM SIGMOD International Conference on Management of Data*, pages 167–178, 2004.

N. de Freitas and H. Kück. Learning about individuals from group statistics. In *Proc. UAI*, pages 332–339, 2005.

L. M. Dery, B. Nachman, F. Rubbo, and A. Schwartzman. Weakly supervised classification in high energy physics. *Journal of High Energy Physics*, 2017(5):1–11, 2017.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. URL http://arxiv.org/abs/1810.04805.

Thomas G. Dietterich, Richard H. Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artif. Intell.*, 89(1-2):31–71, 1997.

Katy Ilonka Gero, Vivian Liu, and Lydia Chilton. Sparks: Inspiration for science writing using language models. In *Proceedings of the 2022 ACM Designing Interactive Systems Conference*, DIS '22, page 1002–1019, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393584. doi: 10.1145/3532106.3533533. URL https://doi.org/10.1145/3532106.3533533.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset, 2021.

Sebastian Hofstätter, Markus Zlabinger, Mete Sertkan, Michael Schröder, and Allan Hanbury. Fine-grained relevance annotations for multi-task document ranking and question answering. In *Proc. of CIKM*, 2020.

Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. True: Re-evaluating factual consistency evaluation. In *Workshop on Document-grounded Dialogue and Conversational Question Answering*, 2022. URL https://api.semanticscholar.org/CorpusID:247694170.

Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/8bb0d291acd4acf06ef112099c16f326-Paper-Conference.pdf.

D. Kotzias, M. Denil, N. de Freitas, and P. Smyth. From group to individual labels using deep features. In *Proc. SIGKDD*, pages 597–606, 2015.

Sayali Kulkarni, Sheide Chammas, Wan Zhu, Fei Sha, and Eugene Ie. Aquamuse: Automatically generating datasets for query-based multi-document summarization, 2020.

Haonan Li, Martin Tomko, Maria Vasardani, and Timothy Baldwin. MultiSpanQA: A dataset for multi-span question answering. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1250–1260, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.90. URL https://aclanthology.org/2022.naacl-main.90.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step, 2023.

J. Liu, B. Wang, Z. Qi, Y. Tian, and Y. Shi. Learning from label proportions with generative adversarial networks. In *Proc. NeurIPS*, pages 7167–7177, 2019.

Jiabin Liu, Bo Wang, Xin Shen, Zhiquan Qi, and Yingjie Tian. Two-stage training for learning from label proportions. In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 2737–2743. International Joint Conferences on Artificial Intelligence Organization, 8 2021. doi: 10.24963/ijcai.2021/377. URL https://doi.org/10.24963/ijcai.2021/377. Main Track.

Jiexi Liu, Dehan Kong, Longtao Huang, Dinghui Mao, and Hui Xue. Multiple instance learning for offensive language detection. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 7387–7396. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.findings-emnlp.546. URL https://aclanthology.org/2022.findings-emnlp.546.

T. Lozano-Pérez and C. Yang. Image database retrieval with multiple-instance learning techniques. In *Proc. ICDE*, page 233, 2000.

Zhekun Luo, Devin Guillory, Baifeng Shi, Wei Ke, Fang Wan, Trevor Darrell, and Huijuan Xu. Weakly-supervised action localization with expectation-maximization multi-instance learning. In *Computer Vision – ECCV 2020*, pages 729–745, Cham, 2020. Springer International Publishing.

O. Maron. *Learning from ambiguity*. PhD thesis, Massassachusetts Institute of Technology, 1998.

Oded Maron and Tomás Lozano-Pérez. A framework for multiple-instance learning. NIPS'97, page 570–576, 1997.

Jacob Menick, Maja Trebacz, Vladimir Mikulik, John Aslanides, Francis Song, Martin Chadwick, Mia Glaese, Susannah Young, Lucy Campbell-Gillingham, Geoffrey Irving, and Nat McAleese. Teaching language models to support answers with verified quotes, 2022. URL https://arxiv.org/abs/2203.11147.

D. R. Musicant, J. M. Christensen, and J. F. Olson. Supervised learning by training on aggregate outputs. In *Proc. ICDM*, pages 252–261. IEEE Computer Society, 2007.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. URL https://arxiv.org/abs/2203.02155.

Nikolaos Pappas and Andrei Popescu-Belis. Explaining the stars: Weighted multiple-instance learning for aspect-based sentiment analysis. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 455–466, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1052. URL https://aclanthology.org/D14-1052.

Nikolaos Pappas and Andrei Popescu-Belis. Explicit document modeling through weighted multiple-instance learning. *Journal of Artificial Intelligence Research*, 58:591–626, 2017.

G. Patrini, R. Nock, T. S. Caetano, and P. Rivera. (almost) no label no cry. In *Proc. Advances in Neural Information Processing Systems*, pages 190–198, 2014.

Laura Perez-Beltrachini, Yang Liu, and Mirella Lapata. Generating summaries with topic templates and structured convolutional decoders. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5107–5116, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1504. URL https://aclanthology.org/P19-1504.

N. Quadrianto, A. J. Smola, T. S. Caetano, and Q. V. Le. Estimating labels from label proportions. *J. Mach. Learn. Res.*, 10:2349–2374, 2009.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2023.

Jan Ramon and Luc De Raedt. Multi instance neural networks. In *Proceedings of the ICML-2000 workshop on attribute-value and relational learning*, pages 53–60, 2000.

Soumya Ray and Mark Craven. Supervised versus multiple instance learning: an empirical comparison. In *Proc. ICML*, page 697–704, 2005.

S. Rueping. SVM classifier estimation from group probabilities. In *Proc. ICML*, pages 911–918, 2010.

Rishi Saket, Aravindan Raghuveer, and Balaraman Ravindran. On combining bags to better learn from label proportions. In *AISTATS*, volume 151 of *Proceedings of Machine Learning Research*, pages 5913–5927. PMLR, 2022. URL https://proceedings.mlr.press/v151/saket22a.html.

C. Scott and J. Zhang. Learning from label proportions: A mutual contamination framework. In *Proc. NeurIPS*, 2020.

Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. Asqa: Factoid questions meet long-form answers, 2023.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. URL https://arxiv.org/abs/2307.09288.

Xi Wang, Fangyao Tang, Hao Chen, Carol Y. Cheung, and Pheng-Ann Heng. Deep semi-supervised multiple instance learning with self-correction for dme classification from oct images. *Medical Image Analysis*, 83:102673, 2023. ISSN 1361-8415. doi: https://doi.org/10.1016/j.media.2022.102673. URL https://www.sciencedirect.com/science/article/pii/S1361841522003012.

J. Wu, Yinan Yu, Chang Huang, and Kai Yu. Deep multiple instance learning for image classification and auto-annotation. In *Proc. CVPR*, pages 3460–3469, 2015.

Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A. Smith, Mari Ostendorf, and Hannaneh Hajishirzi. Fine-grained human feedback gives better rewards for language model training, 2023.

Wenpeng Yin, Dragomir Radev, and Caiming Xiong. Docnli: A large-scale dataset for document-level natural language inference. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.findings-acl.435. URL http://dx.doi.org/10.18653/v1/2021.findings-acl.435.

F. X. Yu, D. Liu, S. Kumar, T. Jebara, and S. F. Chang. $\propto$SVM for learning with label proportions. In *Proc. ICML*, volume 28, pages 504–512, 2013.

Cha Zhang, John Platt, and Paul Viola. Multiple instance boosting for object detection. In *Advances in Neural Information Processing Systems*, volume 18. MIT Press, 2005.

Qi Zhang and Sally Goldman. Em-dd: An improved multiple-instance learning technique. *Advances in neural information processing systems*, 14, 2001.

FRACTAL: Fine-Grained Scoring from Aggregate Text Labels (Appendix)

A Related Work

Multiple Instance Learning (MIL). Here the bag label is modeled as MAX or MIN of the its (unknown) instance-labels (typically labels are $\{0, 1\}$-valued). Introduced to model drug activity detection by Dietterich et al. [1997], MIL has been applied to domains like drug discovery Maron and Lozano-Pérez [1997], time series prediction Maron [1998], retrieval Lozano-Pérez and Yang [2000] and medical imaging Wu et al. [2015].Ramon and De Raedt [2000] proposed a bag-loss method using log-sum exponential approximation to MIN, which was followed by adaptations of boosting and logistic regression Zhang et al. [2005], Ray and Craven [2005], while specialized methods such as EM-DD Zhang and Goldman [2001] have also been developed.

Learning from Label Proportions (LLP). Here the bag-label is the average of the instance-labels, and arises in the context of label privacy concerns Rueping [2010], costly supervision Chen et al. [2004] or lack of labeling instrumentation Dery et al. [2017]. Early work applied traditional supervised learning techniques de Freitas and Kück [2005], Musicant et al. [2007], Rueping [2010], while those of Quadrianto et al. [2009], Patrini et al. [2014] estimated model parameters from bag-labels and Yu et al. [2013] proposed an SVM for LLP. Subsequent work used bag pre-processing Scott and Zhang [2020], Saket et al. [2022] and trained deep networks Kotzias et al. [2015], Liu et al. [2019], specifically Ardehaly and Culotta [2017] proposed the bag-loss method which is commonly used as a baseline.

MIL and LLP for NLP. Applications such as sentiment analysis Pappas and Popescu-Belis [2014], Angelidis and Lapata [2018] and document modeling Pappas and Popescu-Belis [2017] have previously admitted MIL techniques, while more recently Liu et al. [2022] modeled offensive language detection as an MIL problem and proposed a mutual attention based mechanism. On the other hand, the applications of LLP are relatively sparser: Ardehaly and Culotta [2016] applied it to domain adaptation for text data, while recent work Chauhan et al. [2023] proposed a novel method improving on the baseline model training technique of Ardehaly and Culotta [2017] for text classification.

For both MIL and LLP, previous works have proposed pseudo-labeling based model training methods, in which the weak-supervision of bag-labels is used along with model predictions to derive *pseudo-labels* which can be used to train or fine-tune models. For e.g. pseudo-labels are computed via regularization Wang et al. [2023], Liu et al. [2021] or expectation-maximization Luo et al. [2020], Barucic and Kybic [2022] techniques.

B Choice of probAGG

The extension for MIN is: $\mathsf{probAGG}\left(\mathcal{M}(\mathbf{x}_1), \ldots, \mathcal{M}(\mathbf{x}_k)\right) = \mathcal{M}(\mathbf{x}_{i^*})$ where $i^* := \mathsf{argmin}\left(\tilde{y}(\mathbf{x}_1), \ldots, \tilde{y}(\mathbf{x}_k)\right))$ is the index of the instance with the minimum soft-label as defined in Section 3.1. We use the in-built TensorFlow (TF) tf_math_argmin function to compute argmin. For MAX we use the tf_math_argmax function. We investigate other popular differentiable approximations of the MIN function and present experimental results for the binary classification setting in Appendix H. For AVG, we simply take the average of the model predictions i.e., $\mathsf{probAGG}\left(\mathcal{M}(\mathbf{x}_1), \ldots, \mathcal{M}(\mathbf{x}_k)\right) = (1/k) \sum_{i=1}^{k} \mathcal{M}(\mathbf{x}_i)$.

C Pair Prior Loss

The L_{prprior} introduced in Section 3.1 is defined as follows:

$$L_{\text{prprior}}\left(p_{\mathbf{xz}}, \tilde{y}(\mathbf{x}), \tilde{y}(\mathbf{z})\right) := \left| p_{\mathbf{xz}} - \frac{\tilde{y}(\mathbf{x})}{C} \frac{\tilde{y}(\mathbf{z})}{C} \right| \cdot \left| p_{\mathbf{xz}} - \left(1 - \frac{\tilde{y}(\mathbf{x})}{C}\right)\left(1 - \frac{\tilde{y}(\mathbf{z})}{C}\right) \right| \tag{9}$$

We assume that $p_{\mathbf{xz}}$ represents the probability of instances $\mathbf{x}$ and $\mathbf{z}$ having the same label. A low $p_{\mathbf{xz}}$ value indicates that these instances likely belong to different classes. We design a loss function that is minimized when the predicted classes for both instances are consistent with the value of the prior.

D Details on Tasks and Datasets

Additional details for Retrieval and Summarization tasks described in Section 4

Retrieval. We use two datasets for retrieval tasks: MultiSpanQA and FiRA. MultiSpanQA Li et al. [2022] consists of question and retrieved context pairs, with annotated discontinuous answer spans for the train and validation splits (See Tab. 1 for example). We randomly select 25% of the train-split as the test-split. Context is treated as a bag, with sentences as instances labeled 1 if they overlap with annotated spans, and 0 otherwise. The MAX aggregation is used to indicate the presence an answer in the context. Given all MultiSpanQA samples contain answers, we create negative bags for half of the samples by extracting context chunks without answers. Thus, both the instance and bag labels are $\{0, 1\}$-valued.

The FiRA dataset Hofstätter et al. [2020] comprises word-level relevance annotations using $\{0, \ldots, 4\}$-valued labels. We derive the sentence-level scores by taking the word-level average across annotators and then the maximum across all words in a sentence. Similar to the previous setup, we treat the paragraph as a bag, its sentences as instances, and employ MAX as the aggregation function. The instance and bag-level belongs to the set $\{0, 1, 2, 3, 4\}$, with the goal of optimizing a cross entropy loss.

For both datasets, we integrate a correlation prior between sentence pairs and a cosine-similarity prior (see (7), (8)) between the query and each sentence of the context.

Summarization. We utilize two datasets: WikiCatSum Perez-Beltrachini et al. [2019] and AquaMuSe Kulkarni et al. [2020].

We adopt the binary entailment metric for this task. The reference summaries already provided in these two datasets serve as the *entailed* summaries[†] with each sentence considered positively entailed. To generate *non-entailed* summaries, we synthesize negatives similar to Yin et al. [2021]. Firstly, we perturb the reference summary through sentence replacement. This involves randomly selecting k sentences, where k is less than the total sentences in the summary, and iteratively feeding their left context to an PALM-2 Anil et al. [2023] to predict the next sentence. The predicted sentence is then used to replace the selected one. Additionally, we explore the standard word replacement technique, which randomly masks k words whose POS tags are among proper nouns, numbers, and verbs, to introduce factual errors in the summaries. The masked words are then predicted using BERT Devlin et al. [2018]. The perturbed sentences within the summary are considered non-entailed. Thus, the sentence as well as bag labels are $\{0, 1\}$-valued with 1 indicating entailed and 0 non-entailed, with MIN as the aggregation function. Examples of entailed and non-entailed summaries are provided in Appendix L. As in previous tasks, we incorporate sentence-document cosine similarity and sentence correlation priors into our methods. Additionally, we experiment with NLI entailment scores Honovich et al. [2022] as priors for this task.

E Task Specific Discussion of Experimental Results

Following is the detailed per-task analysis:

Long-form Question Answering. As presented in Table 4, the model trained using our PriorsBagLoss method on preference labels with the cosine-similarity prior has the best AUC-ROC, AUC-PR and accuracy scores in the experiments on the QA-feedback dataset. It outperforms both BagLoss as well as the cosine-similarity based baselines, the latter one by a significant margin. However, we observe that the performance of the correlation prior based variant is worse than that of the BagLoss baseline which itself is worse than the Response-level trained baseline.

[†]In this work, we do not filter any noise present in the existing data splits.

Retrieval. In the MultiSpan-QA dataset experiments (Table 4) we observe that the model trained by applying PsLab on the predictions of the model trained on PriorsBagLoss with a combination of the cosine-similarity and correlation priors achieves the best AUC-ROC and AUC-PR scores (by a significant gap) among the bag level baselines, while the variant with only the cosine-similarity prior performs the second bet on these metrics. However, the Response-level trained model and BagLoss achieve marginally higher accuracy scores. All these methods also handily outperform the cosine-similarity baseline. From the FiRA dataset results (Table 5) we observe that the our prior augmented BagLoss method, specifically the using both the priors, performs the best on mae as well as mse metrics.

Summarization. The experimental results on the WikiCatSum dataset, presented in Table 4, show that our PriorsBagLoss method with different combinations of the cosine-similarity and the correlation priors, or using NLI as a prior, as well as the PsLab method on these models yield the best performance (by a significant margin) in terms of AUC-ROC, AUC-PR and accuracy metrics. The comparative baselines are the BagLoss, NLI and cosine-similarity. A similar trend is observed on the AquaMuse dataset (Table 4) on which the our methods significantly outperform the bag-level baselines, in particular the PsLab applied to the PriorsBagLoss method yields the best performing model.

Math Reasoning. On the PRM800k dataset, we observe from the experimental evaluations (Table 4) that the BagLoss and PriorsBagLoss methods are best performing among the bag-level baselines and also outperform the cosine-similarity baselines. While PriorsBagLoss using a combination of cosine-similarity and correlation priors achieves better AUC-ROC scores, BagLoss has significantly better accuracy while the AUC-PR scores are similar.

F Model Training Setup

We use the same model architecture across all tasks: a Sentence-T5 Large encoder to generate embeddings for text components, followed by a 2-hidden layer MLP with 73728 parameters for predicting sentence-level scores. To handle lengthy documents exceeding 2000 tokens in MultiSpanQA, WikiCatSum, and AquaMuSe datasets, we partition documents into 1000-token paragraphs which are encoded separately to improve embedding quality. Subsequently, attention weights representing importance are learnt for each document split, and the document embedding is obtained through a weighted sum of individual split embeddings. We report mean and standard deviation observed over 10 randomly seeded trials. We conduct grid search hyperparameter tuning to identify optimal parameter configurations, including learning rates, weights of prior terms integrated into the loss function, and batch sizes. The range of parameters we searched over and the list of optimal hyperparameters for each dataset is provided in Appendix J.

G Ablation Studies

In Tab. 7 we provide an ablation study on the effectiveness of priors and show how much addition of each prior contributes to the performance of BagLoss shown in Tab. 4. We also adapt our loss functions to ingest instance-level data by adding the instance-level loss term. Tab. 8 shows the performance of the fully supervised model trained on a randomly sampled subset 20% labeled training instances, along with models trained using our methods on the *hybrid* dataset i.e., the 20% labeled training instances along with the remaining bag-level training set. From Tab. 8 we observe that with even 20% instance level labels, we are able to further improve the performance of FRACTAL across all datasets in comparison to Tab. 4, while improving for most metrics on the supervised model trained on 20% of the labeled instances.

H Results for Differentiable Minimum Approximations

In the binary-label case, the standard baseline is Mult which is just the product of the soft-labels. More sophisticated approximations that we include in our study are LSE Ramon and De Raedt [2000], ISR, NOR and GM Zhang et al. [2005] (see Sec. 2.4.1 of Babenko [2008] for details). For the binary case, we can employ the in built TensorFlow (TF) approximation tf_reduce_min over the soft-label (which is used for the loss functions) in our experiments, noting that MAX can be derived from MIN applied to flipped variables in the binary case.

Method	AUC-ROC	AUC-PR	Accuracy
MultiSpanQA			
PriorsBagLoss(0.2, 0)	**0.668 ± 0.054**	**0.313 ± 0.04**	0.836 ± 0.052
PriorsBagLoss(0, 0.2)	0.629 ± 0.055	0.279 ± 0.028	**0.850 ± 0.033**
QA Preference Feedback			
PriosrPrefBagLoss(0.2, 0)	0.517 ± 0.004	0.495 ± 0.003	0.512 ± 0.006
PriorsPrefBagLoss(0, 0.4)	0.528 ± 0.003	0.521 ± 0.002	**0.533 ± 0.004**
PriorsPrefBagLoss(0.2, 0.5)	**0.537 ± 0.003**	**0.531 ± 0.003**	0.519 ± 0.006
WikiCatSum			
PriorsBagLoss(0.2, 0, 0)	0.636 ± 0.019	0.877 ± 0.003	0.639 ± 0.01
PriorsBagLoss(0, 0.3, 0)	0.518 ± 0.005	0.719 ± 0.009	0.391 ± 0.006
PriorsBagLoss(0.2, 0.1, 0)	0.639 ± 0.021	**0.885 ± 0.009**	**0.653 ± 0.013**
PriorsBagLoss(0, 0, 0.4)	**0.643 ± 0.024**	0.881 ± 0.012	0.652 ± 0.017
PRM800K			
PriorsBagLoss(0.6, 0)	0.573 ± 0.014	0.889 ± 0.008	**0.624 ± 0.017**
PriorsBagLoss(0, 0.1)	0.577 ± 0.023	0.925 ± 0.017	0.603 ± 0.038
PriorsBagLoss(0.5, 0.1)	**0.588 ± 0.017**	**0.891 ± 0.006**	0.622 ± 0.015

Table 7: Prior Ablation: Across all tasks, we see that using priors improves performance over all baselines given in Table 4 and that Point,Pair priors have additive benefits.

Method	AUC-ROC	AUC-PR	Accuracy
MultiSpanQA Supervised (20%)	0.658 ± 0.019	0.299 ± 0.013	0.828 ± 0.031
FRACTAL*	**0.691 ± 0.031**	**0.325 ± 0.027**	0.844 ± 0.051
QA Preference Supervised (20%)	0.576 ± 0.007	0.539 ± 0.004	0.603 ± 0.012
FRACTAL**	**0.585 ± 0.015**	**0.54 ± 0.006**	**0.607 ± 0.008**
WikiCatSum Supervised (20%)	**0.773 ± 0.041**	0.88 ± 0.019	0.652 ± 0.039
FRACTAL*	0.662 ± 0.016	0.881 ± 0.009	0.674 ± 0.005
PRM800K Supervised (20%)	0.592 ± 0.012	0.897 ± 0.01	**0.686 ± 0.017**
FRACTAL*	**0.599 ± 0.007**	**0.912 ± 0.004**	0.651 ± 0.009

Table 8: Hybrid learning ablations with 20% instance-level data. * indicates PSLAB method and ** indicates PriorsPrefBagLoss.

Table 9 has an ablation of Mult, GM and tf_reduce_min for the BagLoss and PriorBagLoss methods on the WikiCatSum dataset, demonstrating that tf_reduce_min outperforms the others in AUC-ROC and AUC-PR metrics.

I Aggregate and instance-level evaluations

We also include evaluations of the various methods on a test set of bags w.r.t. bag-level metrics using the corresponding AGG approximations. Tables 10, 11, 12, 13, 14 and 11 contain the aggregate as well as instance evaluations.

Table 16 compares the performance achieved on the WikiCatSum dataset by varying the percentage of 1-bags in the training set.

Model	AND Approx	AUC-ROC	AUC-PR
Instance Baseline	-	0.837	0.894
BagLoss	Mult	0.449	0.785
	GM	0.463	0.824
	tf.reduce_min	0.478	0.829
PriorBagLoss(0.2, 0.1)	Mult	0.599	0.858
	GM	0.631	0.862
	tf.reduce_min	0.643	0.877

Table 9: Results for differentiable AND approximations on WikiCatSum dataset

Evaluation	Model	AUC-ROC	AUC-PR	Accuracy	Precision	Recall
Aggregate	Cosine Similarity	0.488	0.287	0.698	0	0
	NLI	0.6855	0.522	0.7453	0.7883	0.319
	Response-level Model	0.681 ± 0.012	0.525 ± 0.081	0.726 ± 0.033	0.752 ± 0.011	0.428 ± 0.063
	Sentence-level Model	0.653 ± 0.076	0.462 ± 0.050	0.723 ± 0.015	0.599 ± 0.093	0.435 ± 0.187
	BagLoss	0.678 ± 0.082	0.525 ± 0.072	0.718 ± 0.057	0.717 ± 0.014	0.391 ± 0.033
	0.8 BagLoss + 0.2 P1	0.683 ± 0.053	0.527 ± 0.02	0.722 ± 0.035	0.748 ± 0.009	0.461 ± 0.01
	0.7 BagLoss + 0.2 P2 + 0.1 P1	0.665 ± 0.061	0.491 ± 0.04	0.727 ± 0.029	0.693 ± 0.018	0.316 ± 0.037
Instance	Cosine Similarity	0.455	0.135	0.851	0	0
	NLI	0.631	0.366	0.859	0.872	0.178
	Response-level Model	0.583 ± 0.187	0.217 ± 0.094	0.852 ± 0.003	0.529 ± 0.074	0.086 ± 0.04
	Sentence-level Model	0.729 ± 0.016	0.354 ± 0.008	0.861 ± 0.046	0.717 ± 0.011	0.438 ± 0.072
	BagLoss	0.661 ± 0.092	0.309 ± 0.127	0.852 ± 0.133	0.711 ± 0.074	0.24 ± 0.188
	0.8 BagLoss + 0.2 P1	0.669 ± 0.063	0.311 ± 0.059	0.838 ± 0.071	0.65 ± 0.089	0.189 ± 0.112
	0.7 BagLoss + 0.2 P2 + 0.1 P1	0.625 ± 0.07	0.271 ± 0.039	0.851 ± 0.021	0.639 ± 0.189	0.135 ± 0.098
	FGLAB	0.693 ± 0.115	0.326 ± 0.071	0.842 ± 0.052	0.676 ± 0.043	0.228 ± 0.091

Table 10: Comparison of aggregate and instance-level performance on MultiSpanQA Dataset

Evaluation	Method	AUC-ROC	AUC-PR	Accuracy	Precision	Recall
Preference	Cosine Similarity	0.4978	0.3952	0.477	0.379	0.4985
	Response-level Model	0.546	0.4651	0.553	0.437	0.539
	BagLoss	0.543	0.4644	0.5463	0.442	0.5324
	PriorBagLoss(0.2,0)	0.568	0.4658	0.574	0.439	0.5467
Instance	Sentence-level Model	0.647	0.611	0.686	0.722	0.418
	Cosine Similarity	0.535	0.526	0.483	0.893	0.134
	Response-level Model	0.491	0.4643	0.453	0.882	0.278
	BagLoss	0.509	0.5269	0.5167	0.814	0.36
	PriorBagLoss(0.2, 0)	0.516	0.4936	0.508	0.647	0.715

Table 11: Comparison of Preference and instance-level evaluation on QA Preference Feedback Dataset

J Hyperparameter Tuning

The key hyperparameters in our approach include the weights of the bag loss and prior loss terms, as well as the learning rate. We conducted a grid search over various values for these parameters to identify the optimal combination for each dataset. The learning rates considered were $\{1e-2, 1e-3, 1e-4, 1e-5, 1e-6\}$, and the weights for each term were selected from $\{0, 0.1, 0.2, \cdots 1\}$. As noted, we only consider convex combinations of the different loss terms, so combinations where the sum of the coefficients exceeds 1 were excluded. Table 17 contains the best weights for our prior augmented BagLoss method on different datasets, along with the best learning rates and batch size in the bag-level training.

K Details on Test Splits

We use the original test split for all datasets except WikiCatSum, whenever applicable.
PRM800K: We use both Phase 1 and 2 of the PRM800K dataset and maintain the same test splits as the original dataset. We split the train set into 75% for training and 10% for validation.
MultiSpanQA: We randomly select 25% of the train split to form the test split. This is because the

Evaluation	Model	MAE	MSE
Aggregate	Instance-level Model	0.375 ± 0.09	0.247 ± 0.113
	Response-level Model	0.320 ± 0.042	0.197 ± 0.017
	BagLoss	0.326 ± 0.021	0.209 ± 0.007
Instance	Instance-level Model	0.283 ± 0.072	0.141 ± 0.088
	Response-level Model	0.319 ± 0.047	0.186 ± 0.098
	BagLoss	0.304 ± 0.007	0.163 ± 0.002
	PriorBagLoss(0.3, 0)	0.298 ± 0.002	0.157 ± 0.004
	PriorBagLoss(0.2, 0.2)	0.294 ± 0.003	0.155 ± 0.001

Table 12: Comparison of Aggregate and Instance-level evaluations on FirA Dataset

Method	AUC-ROC	AUC-PR	Accuracy	Precision	Recall
Instance Baseline	0.837 ± 0.062	0.894 ± 0.085	0.718 ± 0.085	0.926 ± 0.001	0.733 ± 0.003
NLI	0.639	0.817	0.648	0.834	0.559
Cosine Similarity	0.408	0.829	0.362	0.719	0.276
BagLoss	0.477 ± 0.093	0.831 ± 0.052	0.562 ± 0.047	0.769 ± 0.018	0.319 ± 0.048
PriorBagLoss(0.2, 0.1)	0.641 ± 0.028	0.879 ± 0.013	0.651 ± 0.017	0.897 ± 0.009	0.658 ± 0.082
PSLAB	0.645 ± 0.038	0.879 ± 0.057	0.663 ± 0.091	0.884 ± 0.076	0.661 ± 0.092
0.6*BagLoss + 0.4*NLI	0.642	0.885	0.653	0.914	0.619

Table 13: Instance-level Evaluation on WikiCatSum

Method	AUC-ROC	AUC-PR	Accuracy	Precision	Recall
Sentence-level Model	0.613 ± 0.028	0.928 ± 0.021	0.709 ± 0.051	0.902 ± 0.018	0.993 ± 0.004
Response-level Model	0.528 ± 0.145	0.895 ± 0.037	0.521 ± 0.082	0.851 ± 0.055	0.577 ± 0.065
Cosine Similarity	0.420	0.873	0.496	0.888	0.683
BagLoss	0.569 ± 0.019	0.924 ± 0.011	0.688 ± 0.071	0.904 ± 0.014	0.955 ± 0.048
PriorBagLoss(0.5, 0)	0.582 ± 0.034	0.927 ± 0.009	0.534 ± 0.044	0.920 ± 0.014	0.606 ± 0.037
PriorBagLoss(0, 0.1)	0.579 ± 0.052	0.925 ± 0.017	0.603 ± 0.038	0.908 ± 0.020	0.794 ± 0.055
PriorBagLoss(0.1, 0.1)	0.580 ± 0.068	0.926 ± 0.024	0.563 ± 0.049	0.914 ± 0.028	0.708 ± 0.077
psl	0.597 ± 0.093	0.927 ± 0.004	0.578 ± 0.063	0.911 ± 0.009	0.713 ± 0.128

Table 14: Instance-level Evaluation on PRM800K

original dataset's test split lacks annotated answer spans.
FiRA: We partitioned the samples into 75% for training, 10% for validation, and 15% for testing.
WikiCatSum: We subsample 750 samples from test splits of the Animal and Film domains.
AquaMuSe: The original test split of the abstractive version of AquaMuSe has been utilized for testing purposes.
QA Feedback: Preference annotations were available for both the training and development sets. We reserved the dev set for validation purposes and divided the original training dataset into an 80:20 ratio for training and testing. We only consider wins and losses and have removed any ties before splitting the dataset.

L Generation of Perturbed Summaries from the WikiCatSum and Aquamuse Datasets

We utilize two datasets: WikiCatSum and AquaMuSe for the entailment (or summarization) task. The WikiCatSum dataset Perez-Beltrachini et al. [2019] is specifically designed for multi-document summarization tasks, focusing on generating Wikipedia-style lead sections for entities within three domains: Companies, Films, and Animals out of which we focus on the Films and Animals domains. On the other hand, the AquaMuSe dataset Kulkarni et al. [2020] is tailored for multi-document, question-focused summarization.

We adopt the binary entailment metric for this task. The reference summaries already provided in these two datasets serve as the *entailed* summaries[†]. Each sentence in these summaries is considered positively entailed. To generate *non-entailed* summaries, we synthesize negatives by employing various manipulations, similar to Yin et al. [2021]. Firstly, we perturb the reference summary through sentence replacement. This involves randomly selecting k sentences, where k is less than the total

[†]In this work, we do not filter any noise present in the existing data splits.

Method	AUC-ROC	AUC-PR	Accuracy	Precision	Recall
Sentence-level Model	0.648	0.611	0.686	0.722	0.418
Cosine Similarity	0.535	0.526	0.483	0.893	0.134
Response-level Model	0.491	0.4643	0.453	0.882	0.278
BagLoss	0.509	0.5269	0.5167	0.814	0.36
PriorBagLoss(0.2, 0)	0.516	0.4936	0.508	0.647	0.715
PriorBagLoss(0, 0.4)	0.527	0.515	0.529	0.519	0.763
PriorBagLoss(0.2, 0.5)	0.532	0.522	0.521	0.738	0.469

Table 15: QA Preference Feedback Results (Instance Evaluation)

Method	AUC-ROC	AUC-PR	Accuracy
30% 1-bags			
Supervised	0.735 ± 0.023	0.888 ± 0.006	0.762 ± 0.018
BagLoss	0.605 ± 0.011	0.879 ± 0.006	0.713 ± 0.015
PriorBagLoss(0.2, 0)	0.622 ± 0.018	0.887 ± 0.009	0.724 ± 0.032
40% 1-bags			
Supervised	0.759 ± 0.028	0.887 ± 0.01	0.771 ± 0.015
BagLoss	0.569 ± 0.017	0.866 ± 0.004	0.694 ± 0.027
PriorBagLoss(0.2, 0)	0.618 ± 0.01	0.885 ± 0.006	0.723 ± 0.02
50% 1-bags			
Supervised	0.831 ± 0.051	0.889 ± 0.062	0.714 ± 0.048
BagLoss	0.478 ± 0.065	0.829 ± 0.038	0.569 ± 0.036
PriorBagLoss(0.2, 0.1)	0.639 ± 0.021	0.881 ± 0.009	0.653 ± 0.013
60% 1-bags			
Supervised	0.784 ± 0.02	0.886 ± 0.011	0.738 ± 0.027
BagLoss	0.617 ± 0.021	0.882 ± 0.014	0.697 ± 0.022
PriorBagLoss(0.2, 0)	0.629 ± 0.031	0.883 ± 0.01	0.668 ± 0.013
70% 1-bags			
Supervised	0.721 ± 0.035	0.885 ± 0.013	0.743 ± 0.028
BagLoss	0.563 ± 0.019	0.861 ± 0.011	0.688 ± 0.025
PriorBagLoss(0.2, 0)	0.626 ± 0.017	0.886 ± 0.03	0.724 ± 0.033

Table 16: Comparison of performance achieved on the WikiCatSum dataset by varying the percentage of 1-bags in the training set.

Dataset	α_1	α_2	α_3	Learning Rate	Batch Size
QA-Feedback	0.3	0.2	0.5	1e-5	256
MultiSpanQA	0.8	0.2	0	1e-3	512
WikiCatSum	0.7	0.2	0.1	1e-4	1024
FiRA	0.6	0.2	0.2	1e-5	1024
AquaMuSe	0.7	0.2	0.1	1e-3	512
PRM800K	0.8	0.1	0.1	1e-4	64

Table 17: α_1, α_2, and α_3 represent the coefficients of the BagLoss, cosine similarity prior and correlation prior terms in the loss function.

sentences in the summary, and iteratively feeding their left context to an ULM to predict the next sentence. The predicted sentence is then used to replace the selected one. Additionally, we explore the standard word replacement technique, which randomly masks k words whose POS tags are among proper nouns, numbers, and verbs, to introduce factual errors in the summaries. The masked words are then predicted using BERT. The number of replaced sentences and words is randomly selected for each sample. The perturbed sentences within the summary are considered non-entailed, while the remaining unchanged sentences are deemed entailed. Thus, the sentence as well as bag labels are $\{0, 1\}$-valued with 1 indicating entailed and 0 non-entailed, with MIN as the aggregation function.

Tables 18 and 19 contain the entailed and non-entailed (perturbed) summaries for the Aquamuse and WikiCatSum datasets respectively.

M PsLab for Multiclass

Suppose that the label set is $\{0, \ldots, C\}$ where $C \in \mathbb{Z}^+$ and $C > 1$. With the setup as in Sec. 3.2, here we describe PsLab for MAX aggregation which is used in the FirA dataset. Note that MIN aggregation is equivalent to MAX with the ordering on the labels reversed.

Case $\{0, \ldots, C\}$-labels and MAX aggregation. The algorithm $\mathcal{A}^{\mathsf{MAX}}_{\text{PsLab}}$, on a given bag B and y_B, outputs $\Gamma_B : B \to \{0, \ldots, C\}$ as

1. For each $\mathbf{x} \in B$, let $\Gamma_B(\mathbf{x}) = \text{argmax}_{\ell \in \{0,\ldots,C\}} \mathcal{M}(\mathbf{x})[\ell]$.
2. If $y_B > \max_{\mathbf{x} \in B} \Gamma_B(\mathbf{x})$ then:
   (a) Let $\mathbf{z} := \text{argmax}_{\mathbf{x} \in B} \mathcal{M}(\mathbf{x})[y_B] / \mathcal{M}(\mathbf{x})[\Gamma(\mathbf{x})]$.
   (b) Assign $\Gamma_B(\mathbf{z}) = y_B$.
3. else if $y_B < \max_{\mathbf{x} \in B} \Gamma_B(\mathbf{x})$ then:
   (a) Let $S = \{\mathbf{x} \in B \mid \Gamma_B(\mathbf{x}) > y_B\}$.
   (b) For each $\mathbf{x} \in S$: let $\ell = \text{argmax}_{k \in \{0,\ldots,y_B\}} \mathcal{M}(\mathbf{x})[k] / \mathcal{M}(\mathbf{x})[\Gamma_B(\mathbf{x})]$, and set $\Gamma_B(\mathbf{x}) = \ell$.

Type	Summary
Reference Summary	She also is the first ever woman in Indian History to be nominated as the Rajya Sabha member. She is considered the most important revivalist in the Indian classical dance form of Bharatanatyam from its original' sadhir' style, prevalent amongst the temple dancers, Devadasis, she also worked for the re-establishment of traditional Indian arts and crafts.
Word Replacement	She also is the first ever woman in Indian History to be nominated as the Rajya Sabha Independent member. She is considered the most important revivalist in the Indian classical dance form of Kathak from its Nautch 'sadhir' style, prevalent amongst the temple singers. Furthermore, she also advocated for the re-establishment of traditional Indian arts and crafts.
Sentence Replacement	She also is the first ever woman in Indian History to be nominated as the Rajya Sabha member. She is considered the most important revivalist in the Indian classical dance form of Bharatanatyam from its original' sadhir' style, prevalent amongst the temple dancers. She was also a strong advocate for animal welfare and environmental protection, actively participating in campaigns and legislative efforts throughout her life.

Table 18: Example of the entailed and non-entailed versions of the summary from AquaMuSe Dataset. We either use entailed or non-entailed version.

Type	Summary
Reference Summary	the gold spangle (autographa bractea) is a moth of the family noctuidae . it is found in europe , across western siberia and the altai mountains , the northern caucasus , northern turkey and northern iran . its wingspan is 42 – 50 mm . the forewings are brown and gray with large rhomboid golden marks . the hindwings and body are lighter grayish brown . the moth flies from july to august depending on the location , and migrates long distances . the larvae feed on a wide range of plants including hieracium , tussilago farfara , plantago , crepis paludosa , taraxacum , urtica , lamium , stachys and eupatorium cannabinum .
Word Replacement	the gold spangle (autographa californica) is a moth of the family noctuidae . it is found in western north america , across california and the altai mountains , south dakota and new mexico . its wingspan is 16 - 25 mm . the forewings are blue and gray with silver-white long lateral part and a patch of chestnut brown . the hindwings and body are a grayish tan . the moth flies from march to september depending on the location , and migrates long distances . the larvae feed on a wide range of herbaceous plants including legumes such as fabaceae , alfalfas , peas , taraxacum , urtica , lamium , stachys and eupatorium cannabinum .
Sentence Replacement	the gold spangle (autographa bractea) is a moth of the family noctuidae . it is found in europe , across western siberia and the altai mountains , the northern caucasus , northern turkey and northern iran . its wingspan is 42 – 50 mm . the forewing has an inner line below middle finely golden in color, and the outer one is golden at the inner margin only . the hindwings and body are lighter grayish brown . the moth flies from july to august depending on the location , and migrates long distances . Occupying waste ground, gardens and moorland, this species is widespread and fairly common in the north of Britain .

Table 19: Example of the entailed and non-entailed versions of the summary from WikiCatSum Dataset. We either use entailed or non-entailed version.

(c) If $y_B > \max_{\mathbf{x}\in B} \Gamma_B(\mathbf{x})$ repeat Steps 2(a) and 2(b).

4. Output Γ_B.

Note that $\mathcal{A}^{\mathsf{MAX}}_{\text{PsLab}}$ in the $\{0,1\}$-label case, outputs the all 0s assignment for a bag B if $y_B = 0$, and if $y_B = 1$, it first finds the max likelihood assignment, and if it is all $0s$, then it sets the label of $\mathbf{z}$ to 1 which maximizes $\mathcal{M}(\mathbf{x})[1]/\mathcal{M}(\mathbf{x})[0] = \mathcal{M}(\mathbf{x})[1]/(1-\mathcal{M}(\mathbf{x})[1])$ which is the same as maximizing $\mathcal{M}(\mathbf{x})[1]$. Thus, this is equivalent to the algorithm for MIN aggregation described in Sec. 3.2 by flipping the labels. Therefore, it suffices to prove the correctness of $\mathcal{A}^{\mathsf{MAX}}_{\text{PsLab}}$, as we do in the following lemma. To aid our proof, we shall use the definition of likelihood

$$G(\mathcal{M}, B, \Gamma) := \prod_{\mathbf{x}\in B} \mathcal{M}(\mathbf{x})[\Gamma(\mathbf{x})] \tag{10}$$

for bag B and $\Gamma : B \to \{0, \ldots, C\}$.

Lemma M.1. *For any bag B with aggregate MAX label y_B, the output Γ_B of $\mathcal{A}^{\mathsf{MAX}}_{\text{PsLab}}$ maximizes the likelihood over the set of labellings $\mathcal{P} := \{\omega : B \to \{0, \ldots, C\} \mid \max_{\mathbf{x}\in B} \omega(\mathbf{x}) = y_B\}$ i.e., it satisfies:*

$$G(\mathcal{M}, B, \Gamma_B) = \max_{\omega\in\mathcal{P}} G(\mathcal{M}, B, \omega) \tag{11}$$

Proof. Let us first define another set of labellings $\mathcal{Q} := \{\zeta : B \to \{0, \ldots, C\} \mid \max_{\mathbf{x} \in B} \zeta(\mathbf{x}) \leq y_B\}$. Clearly $\mathcal{P} \subseteq \mathcal{Q}$. We have the following lemma.

Lemma M.2. *Let* $\zeta^* = \text{argmax}_{\zeta \in \mathcal{Q}} G(\mathcal{M}, B, \zeta)$ *such that* $\max_{\mathbf{x} \in B} \zeta^*(\mathbf{x}) < y_B$*, and let* $\mathbf{z} := \text{argmax}_{\mathbf{x} \in B} \mathcal{M}(\mathbf{x})[y_B] / \mathcal{M}(\mathbf{x})[\zeta^*(\mathbf{z})]$*. Then, with* $\overline{\omega}$ *defined as* $\overline{\omega}(\mathbf{z}) = y_B$ *and for all* $\mathbf{x} \in B \setminus \{\mathbf{z}\}, \overline{\omega}(\mathbf{x}) = \zeta^*(\mathbf{x})$*, we have that* $G(\mathcal{M}, B, \overline{\omega}) = \max_{\omega \in \mathcal{P}} G(\mathcal{M}, B, \omega)$.

Proof. Let ω^* be a maximizer of $G(\mathcal{M}, B, \omega)$ in $\mathcal{P}$, i.e. $G(\mathcal{M}, B, \omega^*) = \max_{\omega \in \mathcal{P}} G(\mathcal{M}, B, \omega)$. Now since $\mathcal{P} \subseteq \mathcal{Q}$, we have that $G(\mathcal{M}, B, \zeta^*) \geq G(\mathcal{M}, B, \omega^*)$. Let $\mathbf{z}' \in B$ s.t. $\omega^*(\mathbf{z}') = y_B$ which must exist by the definition of $\mathcal{P}$. Now, for all $\mathbf{x} \in B \setminus \{\mathbf{z}'\}$, $\mathcal{M}(\mathbf{x}, \zeta^*(\mathbf{x})) \geq \mathcal{M}(\mathbf{x}, \omega^*(\mathbf{x}))$, otherwise if there is some $\mathbf{x}$ violating this, then one could increase $G(\mathcal{M}, B, \zeta^*)$ by changing $\zeta^*(\mathbf{x})$ to $\omega^*(\mathbf{x})$. Thus, we can define $\omega' \in \mathcal{P}$ where $\omega'(\mathbf{z}') = \omega^*(\mathbf{z}') = y_B$ and for all $\mathbf{x} \in B \setminus \{\mathbf{z}'\}$, $\omega'(\mathbf{x}) = \zeta^*(\mathbf{x})$, so that $G(\mathcal{M}, B, \omega') \geq G(\mathcal{M}, B, \omega^*)$. Now, in ω' changing the label of $\mathbf{z}'$ to $\zeta^*(\mathbf{z}')$ followed by changing the label of $\mathbf{z}$ (defined in the statement of lemma) to y_B yields $\overline{\omega}$ and by the definition of $\mathbf{z}$, this does not decrease the likelihood, thus completing the proof. □

Using the above lemma, we complete the proof by observing that at the start of Step 2(a), $\Gamma_B = \text{argmax}_{\zeta \in \mathcal{Q}} G(\mathcal{M}, B, \zeta)$ s.t. $\max_{\mathbf{x} \in B} \Gamma_B(\mathbf{x}) < y_B$. This of course is true when Γ_B maximizes the likelihood over all labelings. A simple argument shows that this is also true if Γ_B is obtained from Steps 3(b) followed by 3(c) satisfying the inequality in the latter: in this case Γ_B is obtained from a likelihood maximizer over all labelings, and then for each $\mathbf{x}$ whose label is greater than y_B, its label is changed to the one in $\{0, \ldots, y_B\}$ which has the maximum probability. □

N Limitations

Our method for label calibration and pseudo-labeling works well in classification tasks, leading to better performance. However, applying this technique becomes difficult when dealing with preference feedback. Also, if the bag size becomes very large, there is a risk of the bag-loss functions becoming intractable.