Exploring Ordinality in Text Classification: A Comparative Study of Explicit and Implicit Techniques

Siva Rajesh Kasa*
Amazon, India
kasasiva@amazon.com

Aniket Goel*
IIIT Dehli
aniket20281@iiitd.ac.in

Karan Gupta*
Amazon, India
karaniis@amazon.com

Sumegh Roychowdhury*
Amazon, India
sumegr@amazon.com

Anish Bhanushali
Amazon, India
ankeeb@amazon.com

Nikhil Pattisapu
Amazon, India
npattisa@amazon.com

Prasanna Srinivasa Murthy
Amazon, India
sprsn@amazon.com

Abstract

Ordinal Classification (OC) is a widely encountered challenge in Natural Language Processing (NLP), with applications in various domains such as sentiment analysis, rating prediction, and more. Previous approaches to tackle OC have primarily focused on modifying existing or creating novel loss functions that **explicitly** account for the ordinal nature of labels. However, with the advent of Pretrained Language Models (PLMs), it became possible to tackle ordinality through the **implicit** semantics of the labels as well. This paper provides a comprehensive theoretical and empirical examination of both these approaches. Furthermore, we also offer strategic recommendations regarding the most effective approach to adopt based on specific settings.

1 Introduction

Ordinal classification (OC) is a key task in natural language processing (NLP), with many applications that require ordering or ranking in the output such as Sentiment Analysis (Dang et al., 2020), Rating Prediction (Liu, 2020), Age Group Classification (Sánchez-Hevia et al., 2022), etc. In each of these examples, the output categories have a natural order, making these tasks suited to ordinal classification.

Broadly speaking, OC task can be tackled using one of the following two approaches - **explicit** vs **implicit**. The classical explicit approach relies on tweaking the loss function based on an *explicit notion* of distance between labels and penalising based on the degree of misclassification (Castagnos et al., 2022; Díaz and Marathe, 2019). Alternatively, the second *implicit* approach we propose is based on more recent advancements in language modeling. This approach organically engages the semantics of the labels, thereby harnessing their inherent characteristics for the classification task. Thus, while the former represents a time-tested approach, the latter offers a new way to tackle OC, bringing the richness of semantic information into the classification process. Our paper comprehensively studies the different techniques that fall under both these strategies. The classical approach relies on tweaking Cross Entropy (CE) which is a commonly used loss function for nominal classification (NC) (where the assumption is that the classes are mutually exclusive and have no inherent order or relationship to each other) (Yamasaki, 2022; Castagnos et al., 2022; Díaz and Marathe, 2019). The performance in NC task is usually measured in terms of accuracy-based metrics such F1 scores. While CE is optimal for NC, since it treats all misclassifications as the same, it is sub-optimal for OC. There are several approaches/tweaks proposed to extend CE loss for OC, such as the ones proposed by Díaz and Marathe (2019); Castagnos et al. (2022), which adds a penalty based on the absolute difference in the class rankings. These loss functions are designed such that the more the distance between prediction and ground truth, the more the penalty. The performance of these OC tasks is measured in terms of Mean Squared Error (MSE), Mean Absolute Error (MAE) and Off-by-1 (OB1) accuracy (Castagnos et al., 2022).

Our work fills a crucial gap in the current understanding of these losses used in OC. Prior to our research, a unifying analysis of these different loss based approaches was conspicuously absent, making it challenging to holistically compare and contrast their characteristics and performance. In an attempt to address this lacuna, we evaluate these loss functions for four desirable properties, namely, proper scoring rule, unimodality, convexity and ordinality. We also propose a loss function which exhibits these desirable properties while demonstrating that several of these loss functions can be interpreted as specialized versions of the proposed

*These authors contributed equally to this work

Findings of the Association for Computational Linguistics: ACL 2024, pages 5390–5404

generalized loss function.

Pretrained Language Models (PLMs), which include both encoder models (like BERT (Devlin et al., 2019)) and decoder models (like GPT (Radford et al., 2018a)), offer a robust mechanism to learn the semantics or representations of words. These models are trained on vast amounts of text data, during which they develop an understanding of the context in which words are used. Encoder models like BERT learn to predict a word based on its surrounding context, thereby creating a rich and nuanced understanding of word semantics. On the other hand, decoder models like GPT generate a sequence of words, learning to predict the next word in a sentence, hence understanding semantics in a left-to-right context manner. This learning process enables these models to develop high-dimensional vector representations (embeddings) that capture the underlying semantics of words.

Interestingly, these semantically rich embeddings can be utilized to *implicitly* factor in the ordinality of labels in OC tasks. The numeric vector representation of words inherently carries semantic relationships that can mirror ordinal relationships. For instance, the embeddings of words like "good", "neutral", and "bad" reflect their comparative semantics in the vector space. When used for OC tasks, such as sentiment analysis or rating prediction, these learned embeddings can provide a more natural and effective way to understand and encode the ordinality of labels. The model does not just see these labels as distinct classes, but as points on some latent space, thus allowing a more nuanced approach to such classification tasks.

In our study, we make the following contributions to advance the understanding and application of OC in NLP:

- We conduct a comprehensive analysis of several explicit loss-based approaches for Ordinal Classification (OC), examining them through the lens of useful theoretical properties, such as proper scoring rule, convexity, unimodality and ordinality. Our empirical findings demonstrate that previously proposed techniques for solving OC excel primarily in ordinal metrics while compromising performance on nominal metrics. This analysis has led us to propose a hybrid loss function that achieves a better balance between nominal and ordinal metrics compared to the performance of its individual components.
- Further, we study two general PLM based methodologies (encoder & decoder-based) that implicitly factor in ordinality. These approaches signify a paradigm shift from traditional methods, opening up new avenues in ordinal classification.
- Lastly, we undertake an exhaustive comparison of these explicit and implicit approaches under different scenarios, providing necessary ablations and conclude by offering strategic recommendations on the suitable choice between these approaches in §6.

The rest of the paper is organized as follows: In §2, we discuss the various explicit approaches i.e. review the loss functions and discuss their theoretical properties. Using this as motivation, we also propose a new hybrid loss function and study it's properties. In §3, we discuss the encoder models and show how OC can be approached from an entailment-style modeling perspective. In §4, we discuss the decoder models and how OC can be approached from a next word prediction task perspective. Figure 2 depicts the explicit as well as implicit approaches. In §5, we empirically compare these three methods and give our recommendations.

2 Explicit approach: Loss-functions and Analysis

Loss	PSR	UM	CX	Ord
CE	✓	×	✓	low
OLL	✓	×	✓	high
MLL	✓	×	✓	high
SOFT	×	×	✓	low
EMD	✓	×	✓	low
CORAL	✓	×	✓	low
VS-SL	×	✓	×	low
WKL	✓	×	×	low

Table 1: Various properties satisfied by different loss based approaches. *Notation*: PSR - Proper Scoring Rule, UM - Unimodality, CX - Convexity, Ord - Ordinality

Let $\{(\mathbf{X}_i, y_i)\}_{i=1}^N$ be N independent and identically distributed datapoints containing the input features $\mathbf{X}_i$ and their corresponding labels y_i; where $y_i \in \{1, \ldots, K\}$ where K is the number of classes. The output of the classifier is denoted by $\Phi_\theta(\mathbf{X}_i) = \hat{\mathbf{p}}_i = (\hat{p_{i_1}}, \ldots, \hat{p_{i_K}})$ which is a probability distribution over the K classes. Let $\mathbb{I}(y_i)$ be the one-hot encoding of y_i. The classifier is trained by optimizing the parameters θ such that $\frac{1}{N}\sum_{i=1}^N \mathcal{L}_\theta(\hat{\mathbf{p}}_i, \mathbb{I}(y_i))$ reaches a minimum. In the rest of the paper, for ease of understanding, we omit the indexing with

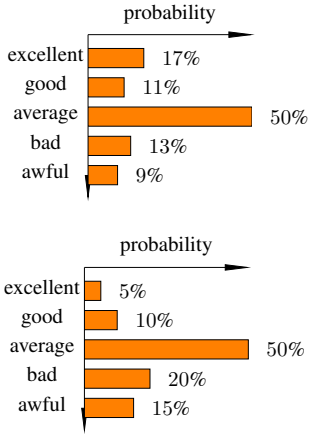

Figure 1: Classifier output on the top does not satisfy UM as the probabilities do not decrease monotonically on either side of the max probability label (*label: average*); whereas the classifier output on the bottom satisfies UM

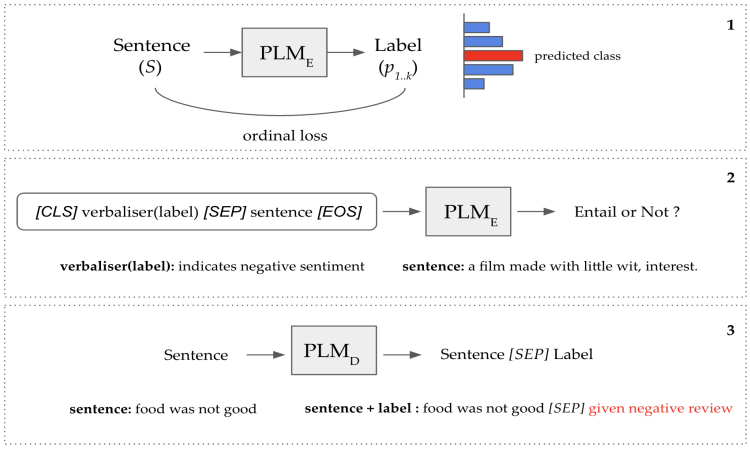

Figure 2: [Best viewed in color] (1) **Explicit Approach:** We explicitly impose ordinality in the predictions by designing losses like MLL, OLL, etc. (2) **Implicit Approach:** In this entailment technique, ordinality is implicitly enforced by passing the label semantic information in the input itself. (3) Here we use a decoder LM to auto-regressively predict the label in natural language format given the input sentence. (*Notation:* PLM_E: Encoder PLM, PLM_D: Decoder PLM)

respect to i, i.e. remove the $\frac{1}{N}$ and $\sum_{i=1}^{N}$, wherever it is evident from the loss expression.

In the next subsection, we give a few desirable theoretical properties of the loss function $\mathcal{L}$ in the context of OC and then follow it up by discussing some of the widely used loss functions.

2.1 Desirable Properties of Losses in OC

Proper Scoring Rule (PSR): A loss is said to be a PSR (Gneiting and Raftery, 2007; Merkle and Steyvers, 2013) if it takes the lowest value when the predicted class probabilities match the ground truth which is a one-hot encoded K-dimensional vector. Being PSR ensures that the loss indeed tries to optimize the classifier to predict the ground truth without injecting any bias in the predicted outputs. Further, PSR losses also help to produce well calibrated probabilities (Lakshminarayanan et al., 2017).

Convexity (Cx): Convexity of $\mathcal{L}$ with respect to $\hat{\mathbf{p}}$ is a desirable property of the loss as it is an essential requirement for several convex formulations of Neural Networks (NNs) (Kawaguchi et al., 2019; Du et al., 2019; Pilanci and Ergen, 2020; Wojtowytsch, 2023). Further, if both $\mathcal{L}$ and Φ (classifier function) are convex with respect to θ (as in the case of logistic regression or support vector machines), then it is guaranteed that the local minima indeed coincides with the global minima.

Several widely used losses such CE, MAE, MSE, etc. are both PSRs and Convex. Next, we look at two desirable characteristics of the output probabilities from the classifier in the context of OC.

Unimodality (UM): If the output probabilities have single mode, i.e. $\hat{p_j} > \hat{p_l} < \hat{p_i}$ is not satisfied for any $1 \leq j < l < i \leq K$, then we say the classifier satisfies the UM condition (Beckham and Pal, 2017; Yamasaki, 2022; Iannario and Piccolo, 2011). An illustration is given in Figure 1.

Ordinality (Ord): In the context of OC, we require the loss $\mathcal{L}$ to explicitly penalize the misclassifications more which are farther away from each other compared to the ones which are closer. The goal is to enforce a meaningful ordering among the labels unlike nominal classifcation where categories lack a specific order. We will see later that different loss functions enforce ordinality to a varying degree.

In the next subsection, we discuss some of the widely used loss functions in the context of OC in NLP.

2.2 Widely used Loss Functions in OC

Cross-Entropy (CE): CE is given by

$$-\sum_{k=1}^{K} p_k \log(\hat{p}_k) \tag{1}$$

The above expression boils down to $-\log(\hat{p}_{y_i})$. Oftentimes CE is discredited for not being able to factor in ordering as its expression does not take into account the probabilities corresponding to the non-groundtruth classes.

Ordinal Log Loss (OLL) (Castagnos et al., 2022): OLL is given by

$$-\sum_{k\neq y}|k-y|^{\alpha}\log(1-\hat{p}_k). \quad (2)$$

Here $\alpha > 0$ is a hyperparameter. This can be seen as a complementary of CE where the misclassification is explicitly penalized in proportion to its degree by the term $|k-y|^{\alpha}$. Both CE and OLL belong to PSR family as both these losses are zero when the output probabilities coincide with the ground truth one-hot encoding.

SOFT labels (SOFT) (Díaz and Marathe, 2019): In this approach, first the ground truth K-dimensional onehot encoded is modified to K 'soft' labels as follows.

$$p_k^{\text{soft}} = \frac{\exp(-\beta|k-y|)}{\sum_{k'}\exp(-\beta|k'-y|)} \quad (3)$$

Then CE loss can be computed using these soft ground labels as truth probabilities.

$$-\sum_{k=1}^{K} p_k^{\text{soft}}\ \log(\hat{p}_k) \quad (4)$$

Clearly, introducing softlabels makes this approach not fall in the PSR family as the ground-truth onehot encoding vector does not minimize the loss anymore.

Earth Mover Distance (EMD) (Rubner et al., 2000) : EMD or Wasserstein loss is defined

$$\text{EMD} = (\text{CDF}(\mathbb{I}(y_i)) - \text{CDF}(\hat{\mathbf{p}}_i))^2$$

Here CDF refers to the cumulative distribution function. While EMD is a PSR, it does not impose a strong penalty in the tails (as compared to OLL) because CDFs are monotonic with range in $[0, 1]$ and hence the difference between the CDFs will be small in the tails.

Unimodal Losses: da Costa et al. (2008) and Beckham and Pal (2017) introduced a parametric way to force unimodality in the predicted probability scores by first computing a scalar function $f(\mathbf{x}) \in [0, 1]$ and then using this scalar to parameterise a Binomial distribution $Bin(f(\mathbf{x}))$. The Probability Mass Function (PMF) is computed which forms the final predicted probabilities. Note that the computing a single scalar $f(x)$ (as opposed a vector of embeddings as done in other methods) severely hampers the learning; thus the guarantee of unimodality comes at the cost of degradation in performance. To address this, Yamasaki (2022) proposes a more-flexible unimodal OC framework by imposing shape-based constraints on the output probabilities; the V-Shaped Stereotyped Logit (VS-SL) method has shown to be the state-of-the-art for UM in their work and hence, we use it as a baseline in our experiments.

We also include two more loss function variants, namely COnsistent RAnk Logits (CORAL) (Cao et al., 2020) and Weighted Kappa Loss (WKL) (de la Torre et al., 2017), and benchmark their effectiveness in terms of accuracy and ordinality.

To benchmark the performance of the above loss functions, we run experiments on three benchmark multi-class text classification datasets, each with distinct tasks: *Hypothesis entails Premise* task - SNLI (Bowman et al., 2015), *Reviews Classification* task - Amazon Reviews (AR) (Keung et al., 2020), and *Sentiment Analysis* task - SST-5 (Socher et al., 2013). We report weighted-F1 scores and MAE, MSE, Off-by-1 (OB1) accuracy to measure both classification and ordinal performance respectively. In the interest of space, we present the details of datasets and metrics in Appendix A and B.

We observe that in general CE performs best in terms of nominal metrics (like weighted-F1) and OLL performs best in terms of ordinal metrics on an average. However, there seems to be a trade-off between nominal and ordinal performance i.e. the improvement in ordinal metrics comes at the expense of nominal metrics. However, given that all these metrics are also from the PSR family, and in some sense are not independent of each other as a perfect classifier would improve both these metrics simultaneously. This intuition prompted us to experiment with a weighted combination of CE and OLL, which we have named Multitask Log Loss (MLL), hypothesizing that it would inherit the best attributes of both methods.

Multi-task log loss function (MLL): MLL is given by

$$MLL = \lambda \times CE + (1-\lambda) \times OLL \quad (5)$$

Here, $\lambda \in [0, 1]$ is a hyperparameter. MLL satisfies both convexity and PSR conditions. Further, while OLL and MLL are not theoretically guaranteed to be UM, but empirically have been found to be satisfying UM condition for 80-90% test datapoints which we show later in Figure 4. A summary of the properties satisfied by different loss-based approaches is given in Table 1, with discussions in Appendix D and E. Note that while it's theoretically possible to consider weighted combinations involving other loss functions, we only considered CE and OLL here because OLL has already been shown to outperform other losses in terms of ordinal metrics (Castagnos et al., 2022) and we aimed

to improve its performance in term of nominal metrics as well, by adding with the CE term.

We follow Castagnos et al. (2022) and using *TinyBERT* as the backbone model, ran CE, OLL, MLL, SOFT, EMD, CORAL, WKL and VS-SL on the three real datasets mentioned in Appendix A and compute the both the nominal and ordinal metrics mentioned in Appendix B. The results are given in Table 2.

Dataset	Loss	LR	F1	MSE	MAE	OB1
SST5	CE	2.5E-05	0.357	1.197	0.768	0.852
			(0.01)	(0.00)	(0.02)	(0.00)
	MLL	1E-04	**0.378**	1.125	0.742	0.863
			(0.02)	(0.01)	(0.01)	(0.00)
	OLL	2.5E-05	0.359	**1.055**	**0.740**	**0.870**
			(0.02)	**(0.00)**	**(0.01)**	**(0.00)**
	WKL	7.5E-05	0.366	1.250	0.809	0.847
			(0.00)	(0.00)	(0.02)	(0.01)
	SOFT	1E-04	0.382	1.152	0.751	0.856
			(0.00)	(0.00)	(0.03)	(0.00)
	EMD	1E-04	0.354	1.125	0.744	0.857
			(0.02)	(0.03)	(0.02)	(0.03)
	CORAL	1E-04	0.109	2.739	1.281	0.641
			(0.00)	(0.01)	(0.06)	(0.00)
	VS_SL	2.5E-05	0.233	2.099	1.087	0.710
			(0.05)	(0.02)	(0.09)	(0.01)
Amazon Reviews	CE	1E-04	0.543	0.904	0.581	0.903
			(0.02)	(0.04)	(0.01)	(0.00)
	MLL	7.5E-05	**0.544**	0.819	**0.568**	0.915
			(0.00)	(0.01)	**(0.00)**	(0.00)
	OLL	1E-04	0.530	**0.788**	0.571	**0.924**
			(0.00)	**(0.00)**	(0.00)	**(0.00)**
	WKL	5E-05	0.515	0.871	0.594	0.907
			(0.00)	(0.00)	(0.01)	(0.05)
	SOFT	7.5E-05	0.537	0.904	0.586	0.903
			(0.00)	(0.00)	(0.01)	(0.00)
	EMD	5E-05	0.534	0.885	0.584	0.904
			(0.00)	(0.00)	(0.00)	(0.00)
	CORAL	1E-04	0.349	1.363	0.781	0.890
			(0.00)	(0.01)	(0.01)	(0.00)
	VS_SL	5E-05	0.377	1.535	0.853	0.793
			(0.05)	(0.08)	(0.03)	(0.02)
SNLI	CE	5E-05	0.821	0.264	0.208	0.972
			(0.00)	(0.00)	(0.02)	(0.01)
	MLL	7.5E-05	**0.832**	0.257	**0.205**	0.974
			(0.00)	(0.01)	**(0.01)**	(0.00)
	OLL	1E-04	0.803	**0.250**	0.217	**0.980**
			(0.01)	**(0.03)**	(0.01)	**(0.02)**
	WKL	1E-04	0.782	0.289	0.242	0.975
			(0.01)	(0.03)	(0.02)	(0.00)
	SOFT	1E-04	0.824	0.257	0.203	0.972
			(0.00)	(0.01)	(0.00)	(0.00)
	EMD	1E-04	0.826	0.250	0.199	0.975
			(0.00)	(0.00)	(0.01)	(0.00)
	CORAL	1E-04	0.815	0.251	0.207	0.977
			(0.00)	(0.02)	(0.02)	(0.01)
	VS_SL	5E-05	0.778	0.339	0.261	0.955
			(0.02)	(0.04)	(0.03)	(0.00)

Table 2: Loss functions comparison on three datasets using BERT-tiny architecture

In order to facilitate fair comparison across explicit and implicit approaches (discussed in the later sections), we repeat the same experiments with *BERT-base-uncased*. The results are given in Table 3 (refer Appendix C for the implementation details). Note that the performance of different loss functions is better contrasted when the size of the base model is small, which is usually the case in online settings where we cannot deploy larger models.

Dataset	Loss	LR	F1	MSE	MAE	OB1
SST5	CE	1E-05	0.484	0.761	0.576	0.925
			(0.01)	(0.04)	(0.02)	(0.01)
	MLL	5E-05	**0.492**	0.757	**0.575**	**0.931**
			(0.01)	(0.01)	**(0.02)**	**(0.00)**
	OLL	5E-05	0.456	**0.735**	0.586	0.927
			(0.01)	**(0.00)**	(0.01)	(0.00)
	WKL	2.5E-05	0.488	0.751	0.584	0.926
			(0.00)	(0.01)	(0.03)	(0.03)
	SOFT	1E-05	0.486	0.761	0.581	0.924
			(0.00)	(0.01)	(0.02)	(0.01)
	EMD	7.5E-05	0.466	0.770	0.592	0.916
			(0.01)	(0.02)	(0.01)	(0.00)
	CORAL	1E-05	0.450	0.927	0.717	0.923
			(0.02)	(0.00)	(0.01)	(0.00)
	VS_SL	1E-05	0.2764	2.663	1.204	0.670
			(0.04)	(0.02)	(0.08)	(0.08)
Amazon Reviews	CE	1E-05	0.586	0.675	0.485	0.938
			(0.04)	(0.01)	(0.06)	(0.03)
	MLL	5E-05	**0.589**	0.634	**0.476**	0.945
			(0.04)	(0.01)	**(0.06)**	(0.03)
	OLL	2.5E-05	0.586	**0.622**	0.477	**0.948**
			(0.00)	**(0.00)**	(0.00)	**(0.00)**
	WKL	1E-05	0.582	0.641	0.482	0.944
			(0.02)	(0.03)	(0.07)	(0.01)
	SOFT	1E-05	0.584	0.681	0.489	0.937
			(0.00)	(0.00)	(0.00)	(0.00)
	EMD	5E-05	0.580	0.652	0.490	0.942
			(0.00)	(0.01)	(0.00)	(0.00)
	CORAL	5E-05	0.406	1.231	0.865	0.966
			(0.00)	(0.01)	(0.01)	(0.00)
	VS_SL	7.5E-05	0.328	1.613	0.907	0.786
			(0.01)	(0.02)	(0.05)	(0.00)
SNLI	CE	5E-05	0.890	0.152	0.123	0.985
			(0.02)	(0.04)	(0.01)	(0.01)
	MLL	5E-05	**0.891**	0.149	0.122	0.986
			(0.02)	(0.04)	(0.02)	(0.01)
	OLL	5E-05	0.890	**0.143**	**0.121**	**0.989**
			(0.01)	**(0.00)**	**(0.01)**	**(0.00)**
	WKL	1E-05	0.865	0.184	0.152	0.984
			(0.01)	(0.01)	(0.00)	(0.00)
	SOFT	5E-05	0.889	0.160	0.127	0.980
			(0.02)	(0.02)	(0.04)	(0.00)
	EMD	1E-05	0.890	0.154	0.126	0.985
			(0.00)	(0.04)	(0.02)	(0.00)
	CORAL	7.5E-05	0.885	0.189	0.128	0.988
			(0.00)	(0.01)	(0.01)	(0.00)
	VS_SL	2.5E-05	0.810	0.300	0.226	0.963
			(0.01)	(0.01)	(0.01)	(0.00)

Table 3: Loss functions comparison on three datasets using BERT-base architecture

3 Implicit approach: Entailment-style Encoder Models

Wang et al. (2021) proposed reformulating vanilla classification into an entailment-style task to enhance the few-shot capabilities of PLMs. Here, the model learns to predict whether the input text and the label entail each other or not (similar to Natural Language Inference (NLI) setting), leveraging the inherent semantic relationship between the label and input text. We adopt a similar approach for our task and explore it through the lens of ordinality, which has not been studied in prior works.

We assume the existence of a classifier, based on Pretrained Language Model (PLM), called Φ_θ. Let $\mathbf{L} = \{L_1, L_2, \ldots, L_K\}$ be the collection of textual labels. The training dataset can be divided based on the labels into $\mathcal{D}_{\text{tr}}$, which consists of subsets $D_1, D_2, \ldots, D_K$. Each subset contains the available training data $\{\mathbf{x}_{k_i}\}_{i=1}^{n_k}$ for label L_k. The corresponding test data is represented by $\mathcal{D}_{\text{tst}}$.

During the training phase in $\mathcal{D}_{\text{tr}}$, the following

entailment-style data augmentation technique is employed: For each data point $\mathbf{x}_{i_j}$ with a ground truth label of L_j, K samples ($\{s^k_{i_j}\}^K_{k=1}$) are generated and augmented as follows:

$$s^k_{i_j} = \{\mathbf{x}_{i_j} + verbaliser(L_k), \mathbb{1}_j(k)\}^K_{k=1} \quad (6)$$

Here, $\mathbb{1}_j(k)$ is an indicator function that yields 1 if $k = j$ else 0. The '+' operator denotes concatenation operation (refer §C), and $verbaliser()$ is a pre-defined template (specific to the downstream task) describing the label in natural language. For example, in sentiment classification task $verbaliser(L_j = positive)$ can be described as: `indicates positive sentiment`. See Figure 2 for example.

Essentially, for each data point, $(K-1)$ negative samples and 1 positive sample are created. Finally the problem reduces to the following NLI task - Does $\mathbf{x}_{i_j}$ entail $verbaliser(L_j)$ or not? Once these $K \times \sum n_k$ augmented examples are generated, the parameters θ are finetuned for a binary classification task, where $s^k_{i_j}$ serves as the input and $\mathbb{1}_j(k)$ acts as the ground truth. During the inference phase, for a datapoint $\mathbf{x}$ the predicted label $\hat{L}$ is obtained using:

$$\hat{L} = \arg\max_k \{\Phi_\theta(s^k)\} \; \forall \; i \in \{1, \ldots, n_k\} \quad (7)$$

During inference for $\mathbf{x}$, s^k is computed following Eq. 6 and `softmax()` is applied on the predicted logits before taking `argmax` so that all the class probabilities sum up to 1. As the model leverages the natural language meanings of the labels during training, we argue it is inherently capable of learning to predict labels that are **ordinally consistent**. For instance, the model learns to comprehend that the label `very negative sentiment` is closer in semantic space to `negative sentiment` than to `very positive sentiment`. This understanding prevents the model from deviating significantly from the actual ground truth. In contrast, in the case of vanilla CE, these labels are treated solely as numbers, disregarding their inter-semantic relations. We again use BERT-base here as base model for performing experiments. The exact label verbalisers used for all datasets are mentioned in Appendix 8.

4 Implicit approach: Generative Models

Decoder-based text generative models have seen notable advancements in recent years, facilitating the production of coherent and contextually relevant text. The development of models like the GPT series (Radford et al., 2018a,b; Brown et al., 2020) has led to state-of-the-art results in text generation and summarization benchmarks. One of the primary objectives of this paper is to investigate whether these models demonstrate ordinal behavior by accurately capturing the **inherent order** or ranking of elements in the generated text.

Formally, in the context of OC, for a given a textual input $\mathbf{x}_i$ which comprises the following words $(w_{i_1}, w_{i_2}, \ldots, w_{i_n})$ and its corresponding ground truth label L_i, we append the input and label as $(w_{i_1}, w_{i_2}, \ldots, w_{i_n}, L_i)$. Next, the parameters θ of generative model are finetuned s.t.

$$\theta^* = \arg\max_\theta \sum_{i=1}^{N} \log P(w_{i_1}, \ldots, w_{i_n}, L_i; \theta)$$

During inference for i^{th} input $\mathbf{x}_i$, the generative model predicts the word $w_{i_{n+1}}$ from the vocabulary $\mathcal{V}$ which maximizes the conditional probability:

$$\hat{L} = \arg\max_{w_{i_{n+1}} \in \mathcal{V}} \{\log P(w_{i_{n+1}} | w_{i_1}, \ldots, ; \theta^*)\} \quad (8)$$

Further, it is possible that the predicted label may not be in the set of ground truth labels i.e. for an i^{th} input $\mathbf{x}_i$ the label $\hat{L}_i \notin \mathbf{L}$ where $\mathbf{L}$ is the set of K distinct labels (see Appendix 9). This is the notorious hallucination problem associated with generative models. To mitigate this issue during inference, we compare the conditional log-probabilities of each the K labels , given the text segment $\mathbf{x}_i$. The class corresponding to the highest probability is proposed as the generated label. That is, given that θ^* represents the learned parameters of LM, the label selection during inference will be s.t.

$$\hat{L} = \arg\max_{L_j \in \mathbf{L}} \log P(L_j | w_{i_1}, \ldots, w_{i_n}; \theta^*) \quad (9)$$

Here $(w_{i_1}, w_{i_2}, \ldots, w_{i_n})$ are the words in input $\mathbf{x}_i$ and $(L_1, L_2, \ldots, L_K)$ are the labels in the set $\mathbf{L}$.

For fair comparison with other explicit and implicit approaches, we use `GPT2-small` as our base model for experiments to maintain similar number of model parameters (with BERT-base). Similar to the entailment approach (§3), we also experiment with informative and un-informative verbalisers here.

Motivated by recent advancements and the accessibility of open-source Large Language Models (LLMs), and to demonstrate the true potential of the generative approach, we also experiment with Llama-7B (Touvron et al., 2023), a decoder-based LLM with 7 billion parameters. Pre-trained on trillions of tokens using publicly available data, it

Dataset	Metric	100 % CE	100 % MLL
SNLI	F1	0.821	0.832
		(0.00)	(0.00)
	MSE	0.264	0.257
		(0.00)	(0.01)
	MAE	0.208	0.205
		(0.02)	(0.01)
	OB1	0.972	0.974
		(0.01)	(0.00)
SST5	F1	0.357	0.378
		(0.01)	(0.02)
	MSE	1.197	1.125
		(0.00)	(0.01)
	MAE	0.768	0.742
		(0.02)	(0.01)
	OB1	0.852	0.863
		(0.00)	(0.00)
Amazon Reviews	F1	0.543	0.544
		(0.02)	(0.00)
	MSE	0.904	0.819
		(0.04)	(0.01)
	MAE	0.581	0.568
		(0.01)	(0.00)
	OB1	0.903	0.915
		(0.00)	(0.00)

Table 4: CE vs MLL using TinyBERT as base model. For full comparison refer Table 2.

		100 %				50 %				25 %				10 %			
		CE	MLL	ENT	GPT	CE	MLL	ENT	GPT	CE	MLL	ENT	GPT	CE	MLL	ENT	GPT
SNLI	F1	0.890	0.891	0.885	0.776	0.873	0.882	0.885	0.654	0.861	0.865	0.869	0.527	0.836	0.848	0.845	0.389
		(0.02)	(0.02)	(0.30)	(0.00)	(0.00)	(0.09)	(0.04)	(0.00)	(0.00)	(0.04)	(0.01)	(0.01)	(0.00)	(0.04)	(0.01)	(0.01)
	MAE	0.123	0.122	0.128	0.261	0.146	0.131	0.129	0.424	0.161	0.153	0.150	0.599	0.191	0.171	0.181	0.811
		(0.01)	(0.02)	(0.00)	(0.00)	(0.00)	(0.01)	(0.00)	(0.00)	(0.00)	(0.01)	(0.00)	(0.02)	(0.00)	(0.05)	(0.00)	(0.01)
	MSE	0.152	0.149	0.157	0.338	0.188	0.156	0.159	0.596	0.206	0.190	0.190	0.878	0.249	0.210	0.236	1.263
		(0.04)	(0.04)	(0.01)	(0.00)	(0.00)	(0.05)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.04)	(0.01)	(0.06)	(0.00)	(0.04)
	OB1	0.985	0.986	0.985	0.961	0.979	0.987	0.980	0.913	0.977	0.981	0.980	0.860	0.971	0.980	0.972	0.774
		(0.01)	(0.01)	(0.02)	(0.00)	(0.00)	(0.02)	(0.00)	(0.00)	(0.00)	(0.02)	(0.00)	(0.01)	(0.00)	(0.00)	(0.00)	(0.01)
SST5	F1	0.484	0.492	0.508	0.487	0.442	0.468	0.46	0.476	0.415	0.440	0.423	0.316	0.417	0.428	0.430	0.274
		(0.01)	(0.01)	(0.01)	(0.02)	(0.02)	(0.03)	(0.02)	(0.00)	(0.03)	(0.04)	(0.03)	(0.01)	(0.02)	(0.08)	(0.03)	(0.02)
	MAE	0.576	0.575	0.543	0.585	0.620	0.600	0.585	0.595	0.652	0.620	0.619	0.966	0.697	0.662	0.651	1.043
		(0.02)	(0.02)	(0.01)	(0.01)	(0.01)	(0.02)	(0.01)	(0.00)	(0.04)	(0.04)	(0.04)	(0.03)	(0.02)	(0.10)	(0.01)	(0.02)
	MSE	0.761	0.757	0.683	0.802	0.845	0.760	0.694	0.819	0.910	0.827	0.870	1.740	1.04	0.905	0.871	1.897
		(0.04)	(0.01)	(0.02)	(0.02)	(0.04)	(0.04)	(0.04)	(0.01)	(0.10)	(0.12)	(0.01)	(0.10)	(0.06)	(0.25)	(0.01)	(0.19)
	OB1	0.925	0.931	0.932	0.918	0.910	0.929	0.93	0.912	0.902	0.918	0.911	0.769	0.875	0.898	0.915	0.734
		(0.01)	(0.00)	(0.01)	(0.00)	(0.08)	(0.01)	(0.01)	(0.00)	(0.01)	(0.02)	(0.02)	(0.01)	(0.01)	(0.05)	(0.02)	(0.01)
Amazon Reviews	F1	0.586	0.589	0.585	0.522	0.573	0.579	0.586	0.449	0.563	0.572	0.573	0.402	0.534	0.553	0.544	0.389
		(0.04)	(0.04)	(0.03)	(0.00)	(0.00)	(0.01)	(0.02)	(0.01)	(0.00)	(0.02)	(0.01)	(0.00)	(0.01)	(0.03)	(0.01)	(0.00)
	MAE	0.485	0.476	0.483	0.573	0.505	0.497	0.480	0.678	0.520	0.503	0.502	0.773	0.56	0.528	0.541	0.813
		(0.06)	(0.06)	(0.01)	(0.01)	(0.00)	(0.01)	(0.00)	(0.00)	(0.00)	(0.07)	(0.02)	(0.00)	(0.01)	(0.05)	(0.02)	(0.00)
	MSE	0.675	0.634	0.664	0.848	0.707	0.683	0.655	1.085	0.739	0.698	0.689	1.348	0.835	0.708	0.767	1.475
		(0.01)	(0.01)	(0.02)	(0.04)	(0.01)	(0.03)	(0.01)	(0.01)	(0.02)	(0.05)	(0.02)	(0.01)	(0.04)	(0.06)	(0.01)	(0.01)
	OB1	0.938	0.945	0.939	0.911	0.932	0.935	0.942	0.864	0.929	0.938	0.935	0.825	0.913	0.934	0.925	0.807
		(0.03)	(0.03)	(0.01)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.03)	(0.01)	(0.00)	(0.00)	(0.02)	(0.01)	(0.00)

Table 5: Comparison of techniques for all 3 datasets in full-data and few-shot settings (100, 50, 25, 10% of data) using BERT-base and GPT2-small as base models. We include one representative loss from each approach - cross-entropy (CE) baseline (nominal), proposed explicit ordinal loss (MLL) and implicit approaches - Entailment (ENT), Generative (GPT).

achieves state-of-the-art performance, surpassing its larger predecessors like GPT-3 (175B) on the majority of benchmarks.

Note that our approach is different from `GPT2ForSquenceClassification`[1] where the last embedding of the last token is used for classification, which is similar to encoder-model (like BERT) style classification. Instead we train it for a language modelling task to generate within a fixed set of tokens i.e. the set of labels.

5 Results and Discussion

For the explicit loss-based approaches, the experimental results presented in Table 4 and 5 show that our proposed hybrid MLL loss improves the ordinal performance (as measured via MAE, MSE, OB1), compared to CE, without compromising its performance along the nominal dimension (as measured via weighted-F1). A detailed comparison of the performance of MLL against other ordinal loss functions is given in Table 3 where we further notice that MLL consistently achieves a balanced performance on both ordinal and nominal metrics, compared to other ordinal losses where nominal metrics degrade in general. But, compared to the results of TinyBERT in Table 2, the difference in performances is less pronounced (Castagnos et al., 2022). The observed phenomenon may be explained by the necessity for the smaller base models to rely more on the explicit enforcement of ordinal losses in enhancing performance in OC tasks. However, as mentioned in §2, given that both the losses and metrics are also from the PSR family, and hence are not independent of each other, a bigger base model would improve both the nominal and ordinal metrics simultaneously. Thus, in offline use cases where it is possible to deploy much bigger models, the difference in performance due to various losses is quite minimal. However, in online settings, where it is imperative to deploy lightweight models due to latency reasons, the difference is quite pronounced.

When considering implicit approaches, we observe that the Entailment (ENT) approach performs **on par** with the MLL loss across all three datasets, without any explicit ordinality enforcing mechanism. It even outperforms CE, MLL, and GPT in almost all data settings for SST-5. This could be attributed to the fact that SST-5 has a significantly lower number of samples (∼12k) compared to SNLI and AR (∼200k). Thus, the ENT approach performs better in few-shot scenarios. Another interesting observation is that for SNLI and AR, although the performances of ENT and MLL are similar, the **standard deviation** numbers are consistently lower in the case of ENT, making it a reliable and relatively more stable approach to use in low-data settings. This supports our claimed hypothesis that the model is able to leverage natural language label descriptions to inherently enforce ordinality in its predictions.

To examine the impact of incorporating informative verbalisers, we also fine-tuned using uninformative ones, replacing the label descriptions

[1] https://tinyurl.com/am93sjdw

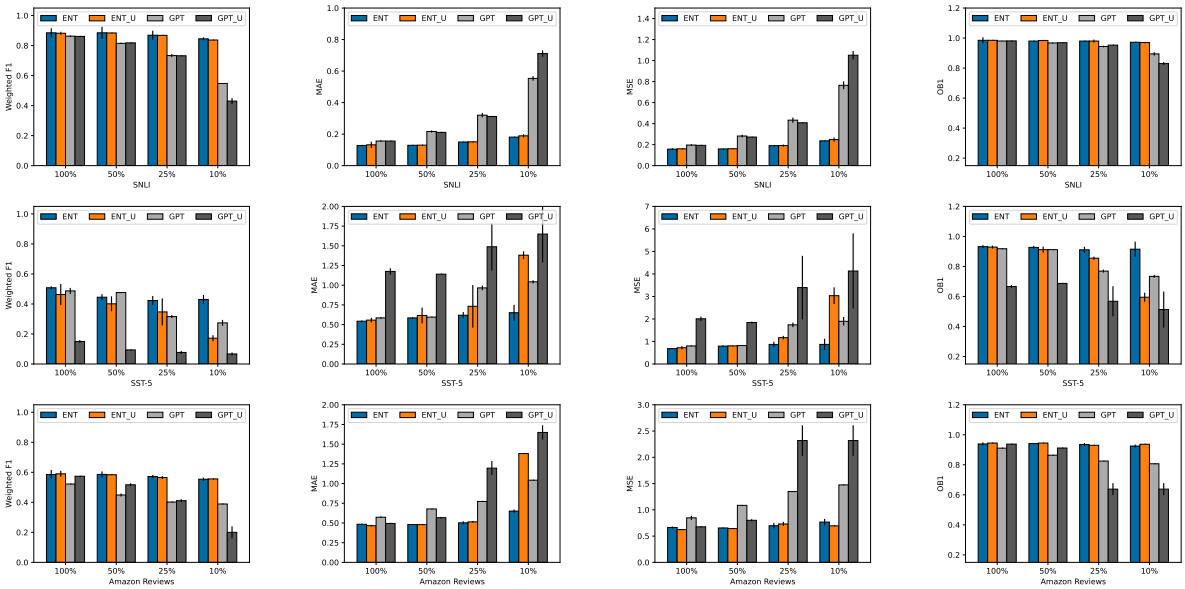

Figure 3: Effect of informative vs uninformative prompts in Implicit approaches on the nominal (F1) and ordinal metrics (MAE, MSE, OB1) on SNLI (*top*), SST5 (*middle*) and Amazon Reviews (*bottom*) datasets. Results averaged across 5 random seeds. (*Notation:* _U refers to the un-informative verbaliser variant. Weighted-F1 and OB1: ↑ better, MAE and MSE: ↓ better.)

with unrelated words such as "cat"/"lion"/"zebra". Our observations, depicted in Figure 3, reveal intriguing findings. When the training data volume is low, the effect of label semantics becomes clearly visible. In the SST-5 dataset (with training data volume ∼10k), the **influence of informative labels** is evident across all settings (refer to *middle* section in Figure 3). Similarly, in the AR and SNLI datasets (with training data volume ∼250k), the impact of informative labels is noticeable only in the 25% or 10% data settings (refer to *top* and *bottom* sections in Figure 3). In other cases, the distinction is less clear due to the sheer volume of training data; even un-informative labels yield decent performance, possibly leveraging some spurious correlations, thus rendering the semantics not so effective. This finding also aligns with the hypothesis proposed by Wang et al. (2021), indicating that prompt-based models may not leverage label semantics as expected, with the dependency being influenced by the model and dataset size to some extent. On another note, for ENT approach, the inference time is scaled by ∼O(K) (K: total labels). But with more carefully engineered verbalisers, similar or even better performance could be achieved even with smaller base models (Jin et al., 2022) making it a suitable candidate for ordinal problems.

In our generative (GEN) approach, we initially employed the GPT2-small model for experiments, as it has a comparable number of fine-tunable parameters to the encoder counterpart used in ENT & Explicit loss approaches. Despite its usual underperformance compared to MLL and ENT, which aligns with the general observation that GPT-2 typically fares worse than BERT-base on most classification tasks (Neerudu et al., 2023), we demonstrate its ability to recognize label order by contrasting informative versus un-informative verbalisers, akin to the ENT approach. Similar trends are observed with the Entailment technique, particularly the significant failure of the un-informative verbaliser variant on the SST-5 dataset. Across other datasets, performance in terms of F1-score is comparable to the informative variant, except for 10% case in SNLI and AR (low-data setting). Notably, for ordinal metrics like MAE and MSE, we observe **high variance** in AR and SST-5 datasets (Figure 3), underscoring the importance of employing informative label verbalizers for stable learning. Two common findings emerge from the informative vs un-informative ablations in both ENT and GEN experiments: **(a)** The hypothesis posited in (Webson and Pavlick, 2022) holds true for ordinal metrics as well as nominal metrics. **(b)** Label semantics become significant only in low-data settings.

The purpose of presenting GPT-2 results was to illustrate the true potential of generative approaches, through LLMs, as introduced in §4. We leverage the *Llama-Adapter* technique (Zhang et al., 2023) which introduces only 1.2M tunable parameters over the base model by prepending a

set of learnable adaption verbalisers to the word tokens at higher transformer layers and show that it still outperforms all explicit & implicit approaches presented above (refer Table 6). While we acknowledge that the enhanced ordinal performance may be attributed to the increased classification performance resulting from **larger model size**, we refer to Table 6 to demonstrate that even with larger base models, the distinction between informative and uninformative verbalisers persists. This underscores the model's ability to recognize label order. In Figure 4 we show the % of samples which follow the unimodality (UM) property empirically. Although there is no theoretical guarantee for both our proposed explicit & implicit strategies, we observe $\sim 95\%+$ **samples** satisfy this property which justifies the increased ordinal performance of the above discussed approaches.

	SST		AR		SNLI	
	I	U	I	U	I	U
F1	**60.845**	58.55	**62.71**	62.4	**90.19**	89.08
	(0.16)	(0.22)	**(0.02)**	(0.1)	**(0.15)**	(0.39)
MAE	**0.4066**	0.4327	**0.4139**	0.4186	**0.1068**	0.1188
	(0.003)	(0.008)	**(0.001)**	(0.003)	**(0.001)**	(0.004)
MSE	**0.4538**	0.4918	**0.5087**	0.5182	**0.1241**	0.1366
	(0.007)	(0.01)	**(0.005)**	(0.008)	**(0.001)**	(0.013)
OB1	**0.9786**	0.9750	**0.963**	0.9595	**0.9913**	0.9911
	(0.001)	(0.003)	**(0.001)**	(0.003)	**(0.001)**	(0.002)

Table 6: Effect of informative (I) vs un-informative (U) verbalisers on Llama-7B

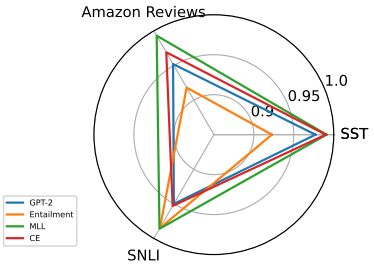

Figure 4: % samples following UM property.

Technique	Recommended Setting
MLL	high-data regime, high % unimodality
ENT	low-data regime, inference time $\sim O(K)$
LLM	high-data regime, compute-intensive, hallucinates in low-data setting

Table 7: Recommended techniques for various settings

6 Conclusion

This paper presents an unified analysis of explicit and implicit strategies for addressing OC. It is the first study to thoroughly examine and compare these approaches from both theoretical and empirical standpoints. Our analysis (summarized in Table 7) reveals that MLL demonstrates balanced performance across ordinal and nominal metrics, unlike existing explicit losses. However, in few-shot scenarios, ENT is preferred for its ability to achieve optimal performance with fewer examples, leveraging label semantics. Furthermore, we highlight the importance of providing informative verbalisers in low-data settings, resulting in reduced variance and improved outcomes. However, the distinction between strategies becomes less clear with increasing data. In full-data scenarios, fine-tuning Llama-7B-Adapter surpasses previous approaches due to its substantial model size. Interestingly, even with such a large base model, the impact of adding informative verbalisers remains apparent, indicating its recognition of label order. We hope that our work will serve as a benchmark encompassing multitude of approaches, providing a foundation for future efforts to address OC in NLP.

7 Limitations

In this study, we don't consider the effect of probability calibration techniques on the explicit approaches (Kull et al., 2019), as the techniques employed are largely identical to those used in nominal classification, offering no distinct or novel methodologies specifically for OC tasks. Also for implicit approaches, a more deeper analysis is required on how implicit methods like LLMs & PLMs implicitly capture ordinality. It's not always analogous to how humans use task instructions as shown in Webson and Pavlick (2022). Furthermore, in this study we limit ourselves to only finetuning-based OC approaches. However, it would also be interesting to explore OC through the lens of in-context learning (ICL) for generative approaches. Also for the generative approach, we make the assumption that the label word will not break further into multiple tokens by re-mapping original labels to simpler words (see Appendix 9). This avoids having to account for multiple token probabilities when taking the argmax. Without this some sort of normalization would be required across the entire generation length to compare different outputs. We leave these discussions for future work. Although we observe that LLMs such as Llama-7B outperform all other models in full-data settings, they pose challenges in terms of compute resources and

inference time. Additionally, fine-tuning Llama-7B is susceptible to hallucinations in low-data settings (Zhao et al., 2021), which is why we do not report LLM results for the few-shot case. Furthermore, the possibility that some of these popular benchmark datasets might have been used for pretraining recent LLMs is an important consideration, not just for this work but for all empirical research involving LLMs. Hence, further research on this issue is warranted.

References

Christopher Beckham and Christopher Pal. 2017. Unimodal probability distributions for deep ordinal classification. In *International Conference on Machine Learning*, pages 411–419. PMLR.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Stephen P Boyd and Lieven Vandenberghe. 2004. *Convex optimization*. Cambridge university press.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Wenzhi Cao, Vahid Mirjalili, and Sebastian Raschka. 2020. Rank consistent ordinal regression for neural networks with application to age estimation. *Pattern Recognition Letters*, 140:325–331.

François Castagnos, Martin Mihelich, and Charles Dognin. 2022. A simple log-based loss function for ordinal text classification. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4604–4609.

Joaquim F Pinto da Costa, Hugo Alonso, and Jaime S Cardoso. 2008. The unimodal model for the classification of ordinal data. *Neural Networks*, 21(1):78–91.

Nhan Cach Dang, María N Moreno-García, and Fernando De la Prieta. 2020. Sentiment analysis based on deep learning: A comparative study. *Electronics*, 9(3):483.

Jordi de la Torre, Domenec Puig, and Aida Valls. 2017. Weighted kappa loss function for multi-class classification of ordinal data in deep learning. *Pattern Recognition Letters*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Simon Du, Jason Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. 2019. Gradient descent finds global minima of deep neural networks. In *International conference on machine learning*, pages 1675–1685. PMLR.

Raúl Díaz and Amit Marathe. 2019. Soft labels for ordinal regression. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4733–4742.

Tilmann Gneiting and Adrian E Raftery. 2007. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378.

Maria Iannario and Domenico Piccolo. 2011. Cub models: Statistical methods and empirical evidence. *Modern Analysis of Customer Surveys: with applications using R*, pages 231–258.

Woojeong Jin, Yu Cheng, Yelong Shen, Weizhu Chen, and Xiang Ren. 2022. A good prompt is worth millions of parameters: Low-resource prompt-based learning for vision-language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2763–2775, Dublin, Ireland. Association for Computational Linguistics.

Kenji Kawaguchi, Jiaoyang Huang, and Leslie Pack Kaelbling. 2019. Every local minimum value is the global minimum value of induced model in nonconvex machine learning. *Neural Computation*, 31(12):2293–2323.

Phillip Keung, Yichao Lu, György Szarvas, and Noah A. Smith. 2020. The multilingual Amazon reviews corpus. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4563–4568, Online. Association for Computational Linguistics.

Meelis Kull, Miquel Perello Nieto, Markus Kängsepp, Telmo Silva Filho, Hao Song, and Peter Flach. 2019. Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30.

Zefang Liu. 2020. Yelp review rating prediction: Machine learning and deep learning models.

Quentin Mérigot, Filippo Santambrogio, and Clément Sarrazin. 2021. Non-asymptotic convergence bounds for wasserstein approximation using point clouds. *Advances in Neural Information Processing Systems*, 34:12810–12821.

Edgar C Merkle and Mark Steyvers. 2013. Choosing a strictly proper scoring rule. *Decision Analysis*, 10(4):292–304.

Pavan Kalyan Reddy Neerudu, Subba Reddy Oota, Mounika Marreddy, Venkateswara Rao Kagita, and Manish Gupta. 2023. On robustness of finetuned transformer-based nlp models. *arXiv preprint arXiv:2305.14453*.

Mert Pilanci and Tolga Ergen. 2020. Neural networks are convex regularizers: Exact polynomial-time convex optimization formulations for two-layer networks. In *International Conference on Machine Learning*, pages 7695–7705. PMLR.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018a. Improving language understanding by generative pre-training. *OpenAI*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2018b. Language models are unsupervised multitask learners.

Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. 2000. The earth mover's distance as a metric for image retrieval. *International journal of computer vision*, 40(2):99.

Héctor A Sánchez-Hevia, Roberto Gil-Pita, Manuel Utrilla-Manso, and Manuel Rosa-Zurera. 2022. Age group classification and gender recognition from speech with temporal convolutional neural networks. *Multimedia Tools and Applications*, 81(3):3535–3552.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. Cite arxiv:2302.13971.

Sinong Wang, Han Fang, Madian Khabsa, Hanzi Mao, and Hao Ma. 2021. Entailment as few-shot learner.

Albert Webson and Ellie Pavlick. 2022. Do prompt-based models really understand the meaning of their prompts? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2300–2344, Seattle, United States. Association for Computational Linguistics.

Stephan Wojtowytsch. 2023. Stochastic gradient descent with noise of machine learning type part i: Discrete time analysis. *Journal of Nonlinear Science*, 33(3):45.

Ryoya Yamasaki. 2022. Unimodal likelihood models for ordinal data. *Transactions on Machine Learning Research*.

Renrui Zhang, Jiaming Han, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, Peng Gao, and Yu Qiao. 2023. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *arXiv preprint arXiv:2303.16199*.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR.

A Datasets

A.1 SNLI

The dataset utilised in this study, initially introduced by Bowman et al. (2015) comprises a substantial collection of 570,000 pairs of human-authored English sentences. This dataset has a CC BY-SA 4.0 license. To ensure robustness, 10,000 pairs have been set aside for both testing and validation purposes. Within this corpus, the labels assigned to each sentence pair are evenly distributed among three distinct categories: entailment, neutral, and contradiction. In order to maintain consistent data sizes across all datasets and expedite the training process, a subset of 250,000 pairs was randomly sampled from the entire collection.

A.2 Amazon Reviews

The dataset utilised in this study, originally introduced by Keung et al. (2020), was constructed by extracting customer reviews from a diverse range of product categories found on the Amazon marketplace. While reviews were collected in six different languages, only those composed in English were retained for training purposes. The dataset is structured for classification tasks and includes the corresponding star ratings, represented as integers spanning from 1 to 5. With a total of 210,000 samples, 5,000 samples were allocated separately for testing and validation purposes. Unlike the SNLI dataset, which underwent a sampling process, the entire dataset was utilised for training in this particular study.

A.3 SST5

The Stanford Sentiment Treebank (SST), originally introduced by Socher et al. (2013), serves as a dedicated corpus tailored specifically for sentiment analysis tasks. This dataset comes with a CC0 (public domain) license. What sets this corpus apart is its incorporation of parse trees, which enable comprehensive sentiment analysis at a granular level. The SST corpus consists of a meticulously curated collection of 12,000 sentences extracted from movie reviews, with each sentence undergoing thorough annotation by three human annotators. In the fine-grained variant of SST, known as SST-5, every individual phrase within the sentences is assigned a rating on a five-star scale. These ratings correspond to distinct sentiment categories, including negative, somewhat negative, neutral, somewhat positive, and positive, providing nuanced insights into the sentiment expressed within the sentences.

Furthermore, for the few-shot learning scenario, we selected subsets consisting of 10%, 25%, and 50% of the data in a randomised manner. This random selection process was repeated for multiple seed values, and the average results were reported to mitigate potential variations.

B Metrics

B.1 Nominal Metrics

B.1.1 F-1 Score

The F1 score is a popular metric for evaluating classification models. It combines precision and recall to provide a single value that represents the model's overall performance. A good F1 score indicates the model's effectiveness in correctly classifying data points.

$$F_1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

B.2 Ordinal Metrics

B.2.1 MSE

Mean Squared Error (MSE), quantifies the average squared difference between the predicted values and the corresponding actual values. A lower MSE signifies a superior alignment between the model's predictions and the ground truth values, thereby indicating heightened accuracy.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

B.2.2 MAE

Mean Absolute Error (MAE), calculates the average absolute difference between the predicted values and the corresponding actual values. MAE offers a straightforward interpretation, representing the average magnitude of errors. A lower MAE value signifies a more accurate alignment between the model's predictions and the ground truth values, indicating a superior fit.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

B.3 Off-by-k Accuracy

Off-by-k accuracy is a metric used to evaluate the performance of a prediction model, particularly in the context of ranking or recommendation systems.

It measures the percentage of predictions that are within k positions of the correct prediction.

$$\text{OB}_k = 100 \times \frac{1}{S} \sum_{s=1}^{S} 1\{d(y_s, \hat{y}_s) \leq k\}$$

C Implementation Details

We employed the *BERT-base-uncased* model as the backbone for our vanilla CE, MLL, and entailment experiments. It consists of a total of 110M parameters including 12 encoder stacks, 12 attention heads, and a hidden state dimension of 768. This model was trained on lower-cased English text. To enable classification, we added a linear layer on top of the backbone model.

For the generative model experiments, we utilized *GPT2-small* as the backbone model, which comprises 117M parameters, including 12 decoder stacks, 12 attention heads, and a hidden state dimension size of 768. We choose the *small* variant to ensure fair comparison with its encoder-model counterpart.

For all the explicit loss-based experiments, we followed the standard supervised learning setup with the necessary modifications to loss function. For the Amazon Reviews (AR), and SST-5 datasets, the input to the model was the text/review and the corresponding class label was provided in a one-hot encoded format. For the SNLI dataset, we passed the input as `Premise [SEP] Hypothesis`.

For the entailment experiments, as mentioned in §3, we included one positive example and (K-1) negative examples. To address data imbalance, for AR, and SST-5, where there are five classes, we augmented an additional positive sample by randomly deleting 5% of the text span in the input text. For SNLI, we created two negative examples corresponding to each positive sample. For all four datasets, we trained for 5 epochs using a learning rate of 5e-5, max sequence length 128, and batch size 32. Through experimentation, we determined the reported settings to yield the best results. Here, the input format to the model was - `verbaliser(label) [SEP] text` for 1-sentence tasks like AR, and SST-5. For 2-sentence task like SNLI, the input format was - `premise [SEP] verbaliser(label) [SEP] hypothesis`.

For the generative model experiments, as mentioned in §4, we trained GPT2 on the language modeling task for 7 epochs using learning rate 1e-5, max sequence length 128, and batch size 32. Positioning the label verbaliser at the end of the text segment demonstrated the best results. The input is in the format - `text [SEP] verbaliser(label)` for 1-sentence tasks and `premise [SEP] hypothesis [SEP] verbaliser(label)` for 2-sentence task. For `verbaliser()` details refer to Table 8 and 9.

For un-informative verbalisers, we used *"cat"/"lion"/"zebra"/"dog"/"snake"* for SST-5, AR and *"cat"/"lion"/"zebra"* for SNLI.

We also conducted additional experiments using the *TinyBERT* model as the backbone to compare various loss functions. The TinyBERT model has a smaller number of 14.5M parameters and is more sensitive to hyperparameters. We trained this model using the same setup as the BERT-base-uncased experiments. The only difference was that we explored different variants of the OLL, MLL, and SOFT loss functions by tuning hyperparameters to compare them with other available loss functions such as CE, EMD, CORAL, etc.

For the Llama-Adapter based experiments, we keep the number of adapter layers at 30, the adaption verbaliser length at 10, max sequence length 512 to accommodate the instructions + verbaliser, batch size 4, weight decay 0.02, base learning rate 9e-3, warmup steps 2 and train it for 5 epochs. The instruction-verbaliser template is kept the same as Alpaca[2] format.

All experiments were conducted with 5 different seeds on 8 Nvidia A100 GPUs in parallel. The reported results include the mean and standard deviations. The CE and MLL experiments took ∼2 hours for AR, and SNLI, while SST-5 required ∼1 hour. In comparison, the entailment experiments took 3-5x longer to train since the dataset size was effectively increased by adding more negative & positive samples. The generative model experiments, took ∼1 hour for AR, and SNLI datasets with max sequence length 128, and ∼30 mins for SST-5. All the reported training times are based on the full-data setting.

D PSR proofs

For a function to be proper scoring rule, it should attain its minimum value at the ground truth. For some losses, like CE and EMD, which are standard losses in ML/NLP, it is well established that they are satisfy by PSR. For others, we verify this property analytically by checking if loss goes to zero when we pass the ground truth information directly.

[2] https://github.com/tloen/alpaca-lora

Dataset	Label Verbaliser
SST-5	*indicates {very negative / negative / neutral / positive / very positive} sentiment*
AR	*given {very negative / negative / neutral / positive / very positive} review*
SNLI	*implies {entailment / neutral / contradiction} to*

Table 8: Verbalisers used for *Entailment-style* approach. The texts inside {} show all the possible label descriptions available to construct the verbaliser.

Dataset	Label Verbaliser
SST-5	*worse / bad / neutral / good / excellent*
AR	*worse / bad / neutral / good / excellent*
SNLI	*yes / fair / no*

Table 9: Verbalisers used for *Generative* approach.

Except for **SOFT** and **VS-SL** rest all other losses shown in Table 1 satisfy this PSR property. Below we give the overall idea on how to check for PSR condition.

Cross-Entropy (CE): It is well established that CE belongs to PSR family (Gneiting and Raftery, 2007).

Ordinal Log Loss (OLL): OLL also belongs to the PSR family since the ground-truth one-hot encoding vector minimizes the loss.

Multi-task log loss function (MLL): Since MLL is a weighted sum of CE and OLL, it can be said that the ground-truth one-hot encoding vector minimizes the MLL loss too since it is already established that both CE and OLL follow this property.

SOFT labels (SOFT): The SOFT loss is similar to CE, computed with the soft ground labels due to which, the one hot encoded ground truth fails to minimise the loss. Hence SOFT loss doesn't belong to the PSR family.

Earth Mover Distance (EMD) : The EMD loss for one particular arbitrary i will be -

$$\text{EMD} = (\text{CDF}(\mathbb{I}(y_i)) - \text{CDF}(\hat{\mathbf{p}}_i))^2$$

When the predicted probabilities $\hat{\mathbf{p}}_i$ coincide with the one hot encoded ground truth $\mathbb{I}(y_i)$, loss value will be zero. Hence EMD does belong to the PSR family.

Weighted Kappa Loss (WKL): The loss is defined as -

$$\kappa = 1 - \frac{\sum_{i,j} w_{ij} O_{ij}}{\sum_{i,j} w_{ij} E_{ij}} \tag{10}$$

where O_{ij} is the observed agreement between the annotators for class i and class j, E_{ij} is the expected agreement by chance, and w_{ij} represents the weight assigned to each class pair.

To define the Weighted Kappa Loss function, we can formulate it as the negative value of the weighted kappa coefficient:

$$L_{\text{weighted\_kappa}} = -\kappa. \tag{11}$$

If our classifier correctly predicts the ground truth our loss becomes zero, which guarantees that WKL is a PSR.

E Convexity proofs

Most of the proofs are one-liners when we employ the standard results of convex optimization. We refer to Boyd and Vandenberghe (2004) for these standard results.

Ordinal Log Loss (OLL): OLL is given by

$$-\sum_{i=1}^{N} \sum_{K:i_k \neq y_i} |i_k - y_i|^{\alpha} \log(1 - \hat{p}_{i_k}). \tag{12}$$

We will use the double derivative approach to prove the convexity. If we can prove the OLL to be convex for any one arbitrary i, then the whole function would also be convex(summation of convex functions is convex), so for an arbitrary i OLL is given by :

$$-\sum_{K:i_k \neq y_i} |i_k - y_i|^{\alpha} \log(1 - \hat{p}_{i_k}). \tag{13}$$

The $|i_k - y_i|^{\alpha}$ term is a positive constant with respect to p_{i_k}, so it won't affect the sign of double derivative and hence can be ignored for easier computations. We now need to check the convexity of just :

$$-\sum_{K:i_k \neq y_i} \log(1 - \hat{p}_{i_k}). \tag{14}$$

To prove the convexity of the given function, we will use the second derivative approach and try to prove that Hessian wrt $\{p_{i_k}\}$ terms is positive semi-

definite (PSD) everywhere on the domain $[0,1]^{K-1}$. For OLL, the Hessian will contain only diagonal terms and if we prove that each of these diagonal terms are positive, then we can say that Hessian is PSD and hence, OLL is convex. The diagonal term corresponding to p_{i_k} is $\frac{1}{(1-p_{i_k})^2}$ which is always non-negative.

MLL: MLL can be viewed as a sum of two functions on the K-dimensional simplex $[0,1]^{K-1}$ i.e.

$$OLL - \log(1 - \sum_{K:i_k \neq y_i} \hat{p}_{i_k}) \tag{15}$$

Previously, we have seen that OLL is convex by proving the Hessian is PSD. Now, we also prove that the Hessian of $-\log(1-\sum_{K:i_k \neq y_i} \hat{p}_{i_k})$ is PSD on $[0,1]^{K-1}$ and hence, by the property that sum of two convex functions on the same domain is convex, we can say MLL is also convex.

It is easy to verify Hessian of $-\log(1 - \sum_{K:i_k \neq y_i} \hat{p}_{i_k})$ is $\frac{1}{(1-\sum_{K:i_k \neq y_i} \hat{p}_{i_k})^2} \mathbb{I}_{K-1 \times K-1}$ where $\mathbb{I}_{K-1 \times K-1}$ is an identity matrix of dimension $K-1 \times K-1$. For the Hessian to be PSD, all diagonal values must be non-negative and it's true in this case. Hence the function $-\log(1 - \sum_{K:i_k \neq y_i} \hat{p}_{i_k})$ is convex along with the OLL being convex in the same domain, which makes MLL a convex loss function.

SOFT labels (SOFT):

The SOFT loss is given by :

$$-\sum_{i=1}^{N} \sum_{k'} p_{i_{k'}}^{\text{soft}} \log(\hat{p}_{i_{k'}}) \tag{16}$$

where -

$$p_{i_k}^{\text{soft}} = \frac{\exp(-\beta |i_k - y_i|)}{\sum_{k'} \exp(-\beta |i_{k'} - y_i|)} \tag{17}$$

The loss function bears resemblance to the Cross-Entropy (CE) loss, with the distinction that the ground truth one-hot encoded label is replaced by softlabels. The term $p_{i_k}^{\text{soft}}$ represents a positive constant concerning the differentiating variable and does not affect the sign of the double derivative. The convexity of the remaining function can be demonstrated in a similar manner as the CE loss. Alternatively, the convex nature of CE implies the convexity of the SOFT loss, and the convexity of CE has been previously established.

Earth Mover Distance (EMD) EMD is also known as Wassertein distance and is well known to be a convex loss on the probabilities (Mérigot et al., 2021).