Understanding Hard Negatives in Noise Contrastive Estimation

Wenzheng Zhang and **Karl Stratos**
Department of Computer Science
Rutgers University
{wenzheng.zhang, karl.stratos}@rutgers.edu

Abstract

The choice of negative examples is important in noise contrastive estimation. Recent works find that hard negatives—highest-scoring incorrect examples under the model—are effective in practice, but they are used without a formal justification. We develop analytical tools to understand the role of hard negatives. Specifically, we view the contrastive loss as a biased estimator of the gradient of the cross-entropy loss, and show both theoretically and empirically that setting the negative distribution to be the model distribution results in bias reduction. We also derive a general form of the score function that unifies various architectures used in text retrieval. By combining hard negatives with appropriate score functions, we obtain strong results on the challenging task of zero-shot entity linking.

1 Introduction

Noise contrastive estimation (NCE) is a widely used approach to large-scale classification and retrieval. It estimates a score function of input-label pairs by a sampled softmax objective: given a correct pair (x, y_1), choose negative examples $y_2 \dots y_K$ and maximize the probability of (x, y_1) in a softmax over the scores of $(x, y_1) \dots (x, y_K)$. NCE has been successful in many applications, including information retrieval (Huang et al., 2013), entity linking (Gillick et al., 2019), and open-domain question answering (Karpukhin et al., 2020).

It is well known that making negatives "hard" can be empirically beneficial. For example, Gillick et al. (2019) propose a hard negative mining strategy in which highest-scoring incorrect labels under the current model are chosen as negatives. Some works even manually include difficult examples based on external information such as a ranking function (Karpukhin et al., 2020) or a knowledge base (Févry et al., 2020).

While it is intuitive that such hard negatives help improve the final model by making the learning task more challenging, they are often used without a formal justification. Existing theoretical results in contrastive learning are not suitable for understanding hard negatives since they focus on unconditional negative distributions (Gutmann and Hyvärinen, 2012; Mnih and Teh, 2012; Ma and Collins, 2018; Tian et al., 2020) or consider a modified loss divergent from practice (Bengio and Senécal, 2008).

In this work, we develop analytical tools to understand the role of hard negatives. We formalize hard-negative NCE with a realistic loss (5) using a general conditional negative distribution, and view it as a biased estimator of the gradient of the cross-entropy loss. We give a simple analysis of the bias (Theorem 3.1). We then consider setting the negative distribution to be the model distribution, which recovers the hard negative mining strategy of Gillick et al. (2019), and show that it yields an unbiased gradient estimator when the model is optimal (Theorem 3.2). We complement the gradient-based perspective with an adversarial formulation (Theorem 3.3).

The choice of architecture to parametrize the score function is another key element in NCE. There is a surge of interest in developing efficient cross-attentional architectures (Humeau et al., 2020; Khattab and Zaharia, 2020; Luan et al., 2020), but they often address different tasks and lack direct comparisons. We give a single algebraic form of the score function (9) that subsumes and generalizes these works, and directly compare a spectrum of architectures it induces.

We present experiments on the challenging task of zero-shot entity linking (Logeswaran et al., 2019). We calculate empirical estimates of the bias of the gradient estimator to verify our analysis, and systematically explore the joint space of negative examples and architectures. We have clear

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1090–1101
June 6–11, 2021.

practical recommendations: (i) hard negative mining always improves performance for all architectures, and (ii) the sum-of-max encoder (Khattab and Zaharia, 2020) yields the best recall in entity retrieval. Our final model combines the sum-of-max retriever with a BERT-based joint reranker to achieve 67.1% unnormalized accuracy: a 4.1% absolute improvement over Wu et al. (2020). We also present complementary experiments on AIDA CoNLL-YAGO (Hoffart et al., 2011) in which we finetune a Wikipedia-pretrained dual encoder with hard-negative NCE and show a 6% absolute improvement in accuracy.

2 Review of NCE

Let $\mathcal{X}$ and $\mathcal{Y}$ denote input and label spaces. We assume $|\mathcal{Y}| < \infty$ for simplicity. Let $\mathbf{pop}$ denote a joint population distribution over $\mathcal{X} \times \mathcal{Y}$. We define a score function $s_\theta : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ differentiable in $\theta \in \mathbb{R}^d$. Given sampling access to $\mathbf{pop}$, we wish to estimate θ such that the classifier $x \mapsto \arg\max_{y \in \mathcal{Y}} s_\theta(x, y)$ (breaking ties arbitrarily) has the optimal expected zero-one loss. We can reduce the problem to conditional density estimation. Given $x \in \mathcal{X}$, define

$$p_\theta(y|x) = \frac{\exp\left(s_\theta(x, y)\right)}{\sum_{y' \in \mathcal{Y}} \exp\left(s_\theta(x, y')\right)} \quad (1)$$

for all $y \in \mathcal{Y}$. Let θ^* denote a minimizer of the cross-entropy loss:

$$J_{\text{CE}}(\theta) = \mathop{\mathbf{E}}_{(x,y)\sim\mathbf{pop}} [-\log p_\theta(y|x)] \quad (2)$$

If the score function is sufficiently expressive, θ^* satisfies $p_{\theta^*}(y|x) = \mathbf{pop}(y|x)$ by the usual property of cross entropy. This implies that s_{θ^*} can be used as an optimal classifier.

The cross-entropy loss is difficult to optimize when $\mathcal{Y}$ is large since the normalization term in (1) is expensive to calculate. In NCE, we dodge this difficulty by subsampling. Given $x \in \mathcal{X}$ and any K labels $y_{1:K} = (y_1 \dots y_K) \in \mathcal{Y}^K$, define

$$\pi_\theta(k|x, y_{1:K}) = \frac{\exp\left(s_\theta(x, y_k)\right)}{\sum_{k'=1}^{K} \exp\left(s_\theta(x, y_{k'})\right)} \quad (3)$$

for all $1 \le k \le K$. When $K \ll |\mathcal{Y}|$, (3) is significantly cheaper to calculate than (1). Given $K \ge 2$, we define

$$J_{\text{NCE}}(\theta) = \mathop{\mathbf{E}}_{\substack{(x,y_1)\sim\mathbf{pop} \\ y_{2:K}\sim q^{K-1}}} [-\log \pi_\theta(1|x, y_{1:K})] \quad (4)$$

where $y_{2:K} \in \mathcal{Y}^{K-1}$ are negative examples drawn iid from some "noise" distribution q over $\mathcal{Y}$. Popular choices of q include the uniform distribution $q(y) = 1/|\mathcal{Y}|$ and the population marginal $q(y) = \mathbf{pop}(y)$.

The NCE loss (4) has been studied extensively. An optimal classifier can be extracted from a minimizer of J_{NCE} (Ma and Collins, 2018); minimizing J_{NCE} can be seen as maximizing a lower bound on the mutual information between $(x, y) \sim \mathbf{pop}$ if q is the population marginal (Oord et al., 2018). We refer to Stratos (2019) for an overview. However, most of these results focus on unconditional negative examples and do not address hard negatives, which are clearly conditional. We now focus on conditional negative distributions, which are more suitable for describing hard negatives.

3 Hard Negatives in NCE

Given $K \ge 2$, we define

$$J_{\text{HARD}}(\theta) = \mathop{\mathbf{E}}_{\substack{(x,y_1)\sim\mathbf{pop} \\ y_{2:K}\sim h(\cdot|x,y_1)}} [-\log \pi_\theta(1|x, y_{1:K})] \quad (5)$$

where $y_{2:K} \in \mathcal{Y}^{K-1}$ are negative examples drawn from a conditional distribution $h(\cdot|x, y_1)$ given $(x, y_1) \sim \mathbf{pop}$. Note that we do not assume $y_{2:K}$ are iid. While simple, this objective captures the essence of using hard negatives in NCE, since the negative examples can arbitrarily condition on the input and the gold (e.g., to be wrong but difficult to distinguish from the gold) and be correlated (e.g., to avoid duplicates).

We give two interpretations of optimizing J_{HARD}. First, we show that the gradient of J_{HARD} is a biased estimator of the gradient of the cross-entropy loss J_{CE}. Thus optimizing J_{HARD} approximates optimizing J_{CE} when we use a gradient-based method, where the error depends on the choice of $h(\cdot|x, y_1)$. Second, we show that the hard negative mining strategy can be recovered by considering an adversarial setting in which $h(\cdot|x, y_1)$ is learned to maximize the loss.

3.1 Gradient Estimation

We assume an arbitrary choice of $h(\cdot|x, y_1)$ and $K \ge 2$. Denote the bias at $\theta \in \mathbb{R}^d$ by

$$b(\theta) = \nabla J_{\text{CE}}(\theta) - \nabla J_{\text{HARD}}(\theta)$$

To analyze the bias, the following quantity will be important. For $x \in \mathcal{X}$ define

$$\gamma_\theta(y|x) = \Pr_{\substack{y_1 \sim \mathbf{pop}(\cdot|x) \\ y_{2:K} \sim h(\cdot|x,y_1) \\ k \sim \pi_\theta(\cdot|x,y_{1:K})}} (y_k = y) \quad (6)$$

for all $y \in \mathcal{Y}$. That is, $\gamma_\theta(y|x)$ is the probability that y is included as a candidate (either as the gold or a negative) and then selected by the NCE discriminator (3).

Theorem 3.1. For all $i = 1 \dots d$,

$$b_i(\theta) = \mathop{\mathbf{E}}_{x \sim \mathbf{pop}} \left[\sum_{y \in \mathcal{Y}} \epsilon_\theta(y|x) \frac{\partial s_\theta(x,y)}{\partial \theta_i} \right]$$

where $\epsilon_\theta(y|x) = p_\theta(y|x) - \gamma_\theta(y|x)$.

Proof. Fix any $x \in \mathcal{X}$ and let $J^x_{\text{CE}}(\theta)$ and $J^x_{\text{HARD}}(\theta)$ denote $J_{\text{CE}}(\theta)$ and $J_{\text{HARD}}(\theta)$ conditioned on x. The difference $J^x_{\text{CE}}(\theta) - J^x_{\text{HARD}}(\theta)$ is

$$\log Z_\theta(x) - \mathop{\mathbf{E}}_{\substack{y_1 \sim \mathbf{pop}(\cdot|x) \\ y_{2:K} \sim h(\cdot|x,y_1)}} [\log Z_\theta(x, y_{1:K})] \quad (7)$$

where we define $Z_\theta(x) = \sum_{y' \in \mathcal{Y}} \exp(s_\theta(x,y'))$ and $Z_\theta(x, y_{1:K}) = \sum_{k=1}^K \exp(s_\theta(x, y_k))$. For any $(\tilde{x}, \tilde{y})$, the partial derivative of (7) with respect to $s_\theta(\tilde{x}, \tilde{y})$ is given by $[[x = \tilde{x}]]\, p_\theta(\tilde{y}|x) - [[x = \tilde{x}]]\, \gamma_\theta(\tilde{y}|x)$ where $[[A]]$ is the indicator function that takes the value 1 if A is true and 0 otherwise. Taking an expectation of their difference over $x \sim \mathbf{pop}$ gives the partial derivative of $b(\theta) = J_{\text{CE}}(\theta) - J_{\text{HARD}}(\theta)$ with respect to $s_\theta(\tilde{x}, \tilde{y})$: $\mathbf{pop}(\tilde{x})(p_\theta(\tilde{y}|\tilde{x}) - \gamma_\theta(\tilde{y}|\tilde{x}))$. The statement follows from the chain rule:

$$b_i(\theta) = \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} \frac{\partial b(\theta)}{\partial s_\theta(x,y)} \frac{\partial s_\theta(x,y)}{\partial \theta_i}$$

□

Theorem 3.1 states that the bias vanishes if $\gamma_\theta(y|x)$ matches $p_\theta(y|x)$. Hard negative mining can be seen as an attempt to minimize the bias by defining $h(\cdot|x, y_1)$ in terms of p_θ. Specifically, we define

$$h(y_{2:K}|x, y_1) \propto [[|\{y_1 \dots y_K\}| = K]] \prod_{k=2}^K p_\theta(y_k|x) \quad (8)$$

Thus $h(\cdot|x, y_1)$ has support only on $y_{2:K} \in \mathcal{Y}^{K-1}$ that are distinct and do not contain the gold. Greedy sampling from $h(\cdot|x, y_1)$ corresponds to taking $K - 1$ incorrect label types with highest scores. This coincides with the hard negative mining strategy of Gillick et al. (2019).

The absence of duplicates in $y_{1:K}$ ensures $J_{\text{CE}}(\theta) = J_{\text{HARD}}(\theta)$ if $K = |\mathcal{Y}|$. This is consistent with (but does not imply) Theorem 3.1 since in this case $\gamma_\theta(y|x) = p_\theta(y|x)$. For general $K < |\mathcal{Y}|$, Theorem 3.1 still gives a precise bias term. To gain a better insight into its behavior, it is helpful to consider a heuristic approximation given by[1]

$$\gamma_\theta(y|x) \approx \frac{p_\theta(y|x) \exp(s_\theta(x,y))}{N_\theta(x)}$$

where $N_\theta(x) = \sum_{y' \in \mathcal{Y}} p_\theta(y'|x) \exp(s_\theta(x,y'))$. Plugging this approximation in Theorem 3.1 we have a simpler equation

$$b_i(\theta) \approx \mathop{\mathbf{E}}_{(x,y) \sim \mathbf{pop}} \left[(1 - \delta_\theta(x,y)) \frac{\partial s_\theta(x,y)}{\partial \theta_i} \right]$$

where $\delta_\theta(x,y) = \exp(s_\theta(x,y)) / N_\theta(x)$. The expression suggests that the bias becomes smaller as the model improves since $p_\theta(\cdot|x) \approx \mathbf{pop}(\cdot|x)$ implies $\delta_\theta(x,y) \approx 1$ where $(x,y) \sim \mathbf{pop}$.

We can formalize the heuristic argument to prove a desirable property of (5): the gradient is unbiased if θ satisfies $p_\theta(y|x) = \mathbf{pop}(y|x)$, assuming iid hard negatives.

Theorem 3.2. Assume $K \geq 2$ and the distribution $h(y_{2:K}|x, y_1) = \prod_{k=2}^K p_\theta(y_k|x)$ in (5). If $p_\theta(y|x) = \mathbf{pop}(y|x)$, then $\nabla J_{\text{HARD}}(\theta) = \nabla J_{\text{CE}}(\theta)$.

Proof. Since $\mathbf{pop}(y|x) = \exp(s_\theta(x,y))/Z_\theta(x)$, the probability $\gamma_\theta(y|x)$ in (6) is

$$\sum_{y_{1:K} \in \mathcal{Y}^K} \prod_{k=1}^K \frac{\exp(s_\theta(x,y_k))}{Z_\theta(x)} \frac{\exp(s_\theta(x,y))}{Z_\theta(x, y_{1:K})}$$

$$= \frac{\exp(s_\theta(x,y))}{Z_\theta(x)} \sum_{y_{1:K} \in \mathcal{Y}^K} \frac{\prod_{k=1}^K \exp(s_\theta(x,y_k))}{Z_\theta(x, y_{1:K})}$$

The sum marginalizes a product distribution over $y_{1:K}$, thus equals one. Hence $\gamma_\theta(y|x) = p_\theta(y|x)$. The statement follows from Theorem 3.1. □

[1] We can rewrite $\gamma_\theta(y|x)$ as

$$\mathop{\mathbf{E}}_{\substack{y_1 \sim \mathbf{pop}(\cdot|x) \\ y_{2:K} \sim h(\cdot|x,y_1)}} \left[\frac{\text{count}_{y_{1:K}}(y) \exp(s_\theta(x,y))}{\sum_{y' \in \mathcal{Y}} \text{count}_{y_{1:K}}(y') \exp(s_\theta(x,y'))} \right]$$

where $\text{count}_{y_{1:K}}(y)$ is the number of times y appears in $y_{1:K}$. The approximation uses $\text{count}_{y_{1:K}}(y) \approx p_\theta(y|x)$ under (8).

The proof exploits the fact that negative examples are drawn from the model and does not generally hold for other negative distributions (e.g., uniformly random). We empirically verify that hard negatives indeed yield a drastically smaller bias compared to random negatives (Section 6.4).

3.2 Adversarial Learning

We complement the bias-based view of hard negatives with an adversarial view. We generalize (5) and define

$$J_{\text{ADV}}(\theta, h) = \mathop{\mathbf{E}}_{\substack{(x,y_1)\sim\mathbf{pop} \\ y_{2:K}\sim h(\cdot|x,y_1)}} [-\log \pi_\theta(1|x, y_{1:K})]$$

where we additionally consider the choice of a hard-negative distribution. The premise of adversarial learning is that it is beneficial for θ to consider the worst-case scenario when minimizing this loss. This motivates a nested optimization problem:

$$\min_{\theta\in\mathbb{R}^d} \max_{h\in\mathcal{H}} J_{\text{ADV}}(\theta, h)$$

where $\mathcal{H}$ denotes the class of conditional distributions over $S \subset \mathcal{Y}$ satisfying $|S \cup \{y_1\}| = K$.

Theorem 3.3. Fix $\theta \in \mathbb{R}^d$. For any (x, y_1), pick

$$\tilde{y}_{2:K} \in \mathop{\arg\max}_{\substack{y_{2:K}\in\mathcal{Y}^{K-1}: \\ |\{y_1 \dots y_K\}|=K}} \sum_{k=2}^{K} s_\theta(x, y_k)$$

breaking ties arbitrarily, and define the point-mass distribution over $\mathcal{Y}^{K-1}$:

$$\tilde{h}(y_{2:K}|x, y_1) = [[y_k = \tilde{y}_k \; \forall k = 2 \dots K]]$$

Then $\tilde{h} \in \arg\max_{h\in\mathcal{H}} J_{\text{ADV}}(\theta, h)$.

Proof. $\max_{h\in\mathcal{H}} J_{\text{ADV}}(\theta, h)$ is equivalent to

$$\max_{h\in\mathcal{H}} \mathop{\mathbf{E}}_{\substack{(x,y_1)\sim\mathbf{pop} \\ y_{2:K}\sim h(\cdot|x,y_1)}} \left[\log \sum_{k=1}^{K} \exp\left(s_\theta(x, y_k)\right)\right]$$

The expression inside the expectation is maximized by $\tilde{y}_{2:K}$ by the monotonicity of log and exp, subject to the constraint that $|\{y_1 \dots y_K\}| = K$. $\tilde{h} \in \mathcal{H}$ achieves this maximum. □

4 Score Function

Along with the choice of negatives, the choice of the score function $s_\theta : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ is a critical component of NCE in practice. There is a clear trade-off between performance and efficiency in modeling the cross interaction between the input-label pair (x, y). This trade-off spurred many recent works to propose various architectures in search of a sweet spot (Humeau et al., 2020; Luan et al., 2020), but they are developed in isolation of one another and difficult to compare. In this section, we give a general algebraic form of the score function that subsumes many of the existing works as special cases.

4.1 General Form

We focus on the standard setting in NLP in which $x \in \mathcal{V}^T$ and $y \in \mathcal{V}^{T'}$ are sequences of tokens in a vocabulary $\mathcal{V}$. Let $E(x) \in \mathbb{R}^{H\times T}$ and $F(y) \in \mathbb{R}^{H\times T'}$ denote their encodings, typically obtained from the final layers of separate pretrained transformers like BERT (Devlin et al., 2019). We follow the convention popularized by BERT and assume the first token is a special symbol (i.e., [CLS]), so that $E_1(x)$ and $F_1(y)$ represent single-vector summaries of x and y. We have the following design choices:

- **Direction**: If $x \to y$, define the query $Q = E(x)$ and key $K = F(y)$. If $y \to x$, define the query $Q = F(y)$ and key $K = E(x)$.
- **Reduction**: Given integers m, m', reduce the number of columns in Q and K to obtain $Q_m \in \mathbb{R}^{H\times m}$ and $K_{m'} \in \mathbb{R}^{H\times m'}$. We can simply select leftmost columns, or introduce an additional layer to perform the reduction.
- **Attention**: Choose a column-wise attention $\text{Attn} : A \mapsto \bar{A}$ either Soft or Hard. If Soft, $\bar{A}_t = \text{softmax}(A_t)$ where the subscript denotes the column index. If Hard, $\bar{A}_t$ is a vector of zeros with exactly one 1 at index $\arg\max_i [A_t]_i$.

Given the design choices, we define the score of (x, y) as

$$s_\theta(x, y) = 1_m^\top Q_m^\top K_{m'} \text{Attn}\left(K_{m'}^\top Q_m\right) \quad (9)$$

where 1_m is a vector of m 1s that aggregates query scores. Note that the query embeddings Q_m double as the value embeddings. The parameter vector $\theta \in \mathbb{R}^d$ denotes the parameters of the encoders E, F and the optional reduction layer.

4.2 Examples

Dual encoder. Choose either direction $x \to y$ or $y \to x$. Select the leftmost $m = m' = 1$ vectors in Q and K as the query and key. The choice of attention has no effect. This recovers the standard dual

encoder used in many retrieval problems (Gupta et al., 2017; Lee et al., 2019; Logeswaran et al., 2019; Wu et al., 2020; Karpukhin et al., 2020; Guu et al., 2020): $s_\theta(x, y) = E_1(x)^\top F_1(y)$.

Poly-encoder. Choose the direction $y \to x$. Select the leftmost $m = 1$ vector in $F(y)$ as the query. Choose an integer m' and compute $K_{m'} = E(x)\text{Soft}(E(x)^\top O)$ where $O \in \mathbb{R}^{H \times m'}$ is a learnable parameter ("code" embeddings). Choose soft attention. This recovers the poly-encoder (Humeau et al., 2020): $s_\theta(x, y) = F_1(y)^\top C_{m'}(x, y)$ where $C_{m'}(x, y) = K_{m'}\text{Soft}\left(K_{m'}^\top F_1(y)\right)$. Similar architectures without length reduction have been used in previous works, for instance the neural attention model of Ganea and Hofmann (2017).

Sum-of-max. Choose the direction $x \to y$. Select all $m = T$ and $m' = T'$ vectors in $E(x)$ and $F(y)$ as the query and key. Choose $\text{Attn} = \text{Hard}$. This recovers the sum-of-max encoder (aka., ColBERT) (Khattab and Zaharia, 2020): $s_\theta(x, y) = \sum_{t=1}^{T} \max_{t'=1}^{T'} E_t(x)^\top F_{t'}(y)$.

Multi-vector. Choose the direction $x \to y$. Select the leftmost $m = 1$ and $m' = 8$ vectors in $E(x)$ and $F(y)$ as the query and key. Choose $\text{Attn} = \text{Hard}$. This recovers the multi-vector encoder (Luan et al., 2020): $s_\theta(x, y) = \max_{t'=1}^{m'} E_1(x)^\top F_{t'}(y)$. It reduces computation to fast dot products over cached embeddings, but is less expressive than the sum-of-max.

The abstraction (9) is useful because it generates a spectrum of architectures as well as unifying existing ones. For instance, it is natural to ask if we can further improve the poly-encoder by using $m > 1$ query vectors. We explore these questions in experiments.

5 Related Work

We discuss related work to better contextualize our contributions. There is a body of work on developing unbiased estimators of the population distribution by modifying NCE. The modifications include learning the normalization term as a model parameter (Gutmann and Hyvärinen, 2012; Mnih and Teh, 2012) and using a bias-corrected score function (Ma and Collins, 2018). However, they assume unconditional negative distributions and do not explain the benefit of hard negatives in NCE (Gillick et al., 2019; Wu et al., 2020; Karpukhin et al., 2020; Févry et al., 2020). In contrast, we directly consider the hard-negative NCE loss used in practice (5), and justify it as a biased estimator of the gradient of the cross-entropy loss.

Our work is closely related to prior works on estimating the gradient of the cross-entropy loss, again by modifying NCE. They assume the following loss (Bengio and Senécal, 2008), which we will denote by $J_{\text{PRIOR}}(\theta)$:

$$\mathop{\mathbf{E}}_{\substack{(x,y_1)\sim\mathbf{pop} \\ y_{2:K}\sim\nu(\cdot|x,y_1)^K}} \left[-\log \frac{\exp\left(\bar{s}_\theta(x, y_1, y_1)\right)}{\sum_{k=1}^{K} \exp\left(\bar{s}_\theta(x, y_1, y_k)\right)} \right] \tag{10}$$

Here, $\nu(\cdot|x, y_1)$ is a conditional distribution over $\mathcal{Y}\backslash\{y_1\}$, and $\bar{s}_\theta(x, y', y)$ is equal to $s_\theta(x, y)$ if $y = y'$ and $s_\theta(x, y) - \log((K-1)\nu(y|x, y_1))$ otherwise. It can be shown that $\nabla J_{\text{PRIOR}}(\theta) = \nabla J_{\text{CE}}(\theta)$ iff $\nu(y|x, y_1) \propto \exp(s_\theta(x, y))$ for all $y \in \mathcal{Y}\backslash\{y_1\}$ (Blanc and Rendle, 2018). However, (10) requires adjusting the score function and iid negative examples, thus less aligned with practice than (5). The bias analysis of $\nabla J_{\text{PRIOR}}(\theta)$ for general $\nu(\cdot|x, y_1)$ is also significantly more complicated than Theorem 3.1 (Rawat et al., 2019).

There is a great deal of recent work on unsupervised contrastive learning of image embeddings in computer vision (Oord et al., 2018; Hjelm et al., 2019; Chen et al., 2020, *inter alia*). Here, $s_\theta(x, y) = E_\theta(x)^\top F_\theta(y)$ is a similarity score between images, and E_θ or F_θ is used to produce useful image representations for downstream tasks. The model is again learned by (4) where (x, y_1) are two random corruptions of the same image and $y_{2:K}$ are different images. Robinson et al. (2021) propose a hard negative distribution in this setting and analyze the behavior of learned embeddings under that distribution. In contrast, our setting is large-scale supervised classification, such as entity linking, and our analysis is concerned with NCE with general hard negative distributions.

In a recent work, Xiong et al. (2021) consider contrastive learning for text retrieval with hard negatives obtained globally from the whole data with asynchronous updates, as we do in our experiments. They use the framework of importance sampling to argue that hard negatives yield gradients with larger norm, thus smaller variance and faster convergence. However, their argument does not imply our theorems. They also assume a pairwise loss, excluding non-pairwise losses such as (4).

6 Experiments

We now study empirical aspects of the hard-negative NCE (Section 3) and the spectrum of score functions (Section 4). Our main testbed is Zeshel (Logeswaran et al., 2019), a challenging dataset for zero-shot entity linking. We also present complementary experiments on AIDA CoNLL-YAGO (Hoffart et al., 2011).[2]

6.1 Task

Zeshel contains 16 domains (fictional worlds like *Star Wars*) partitioned to 8 training and 4 validation and test domains. Each domain has tens of thousands of entities along with their textual descriptions, which contain references to other entities in the domain and double as labeled mentions. The input x is a contextual mention and the label y is the description of the referenced entity. A score function $s_\theta(x, y)$ is learned in the training domains and applied to a new domain for classification and retrieval. Thus the model must read descriptions of unseen entities and still make correct predictions.

We follow prior works and report micro-averaged top-64 recall and macro-averaged accuracy for evaluation. The original Zeshel paper (Logeswaran et al., 2019) distinguishes normalized vs unnormalized accuracy. Normalized accuracy assumes the presence of an external retriever and considers a mention only if its gold entity is included in top-64 candidates from the retriever. In this case, the problem is reduced to reranking and a computationally expensive joint encoder can be used. Unnormalized accuracy considers all mentions. Our goal is to improve unnormalized accuracy.

Logeswaran et al. (2019) use BM25 for retrieval, which upper bounds unnormalized accuracy by its poor recall (first row of Table 1). Wu et al. (2020) propose a two-stage approach in which a dual encoder is trained by hard-negative NCE and held fixed, then a BERT-based joint encoder is trained to rerank the candidates retrieved by the dual encoder. This approach gives considerable improvement in unnormalized accuracy, primarily due to the better recall of a trained dual encoder over BM25 (second row of Table 1). We show that we can further push the recall by optimizing the choice of hard negatives and architectures.

[2]Our code is available at: https://github.com/WenzhengZhang/hard-nce-el.

6.2 Architectures

We represent x and y as length-128 wordpiece sequences where the leftmost token is the special symbol [CLS]; we mark the boundaries of a mention span in x with special symbols. We use two independent BERT-bases to calculate mention embeddings $E(x) \in \mathbb{R}^{768\times128}$ and entity embeddings $F(y) \in \mathbb{R}^{768\times128}$, where the columns $E_t(x), F_t(y)$ are contextual embeddings of the t-th tokens.

Retriever. The retriever defines $s_\theta(x, y)$, the score between a mention x and an entity y, by one of the architectures described in Section 4.2:

$$E_1(x)^\top F_1(y) \qquad \text{(DUAL)}$$
$$F_1(y)^\top C_m(x, y) \qquad \text{(POLY-}m\text{)}$$
$$\max_{t=1}^{m} E_1(x)^\top F_t(y) \qquad \text{(MULTI-}m\text{)}$$
$$\sum_{t=1}^{128} \max_{t'=1}^{128} E_t(x)^\top F_{t'}(y) \qquad \text{(SOM)}$$

denoting the dual encoder, the poly-encoder (Humeau et al., 2020), the multi-vector encoder (Luan et al., 2020), and the sum-of-max encoder (Khattab and Zaharia, 2020). These architectures are sufficiently efficient to calculate $s_\theta(x, y)$ for all entities y in training domains for each mention x. This efficiency is necessary for sampling hard negatives during training and retrieving candidates at test time.

Reranker. The reranker defines $s_\theta(x, y) = w^\top E_1(x, y) + b$ where $E(x, y) \in \mathbb{R}^{H\times256}$ is BERT (either base $H = 768$ or large $H = 1024$) embeddings of the concatenation of x and y separated by the special symbol [SEP], and w, b are parameters of a linear layer. We denote this encoder by JOINT.

6.3 Optimization

Training a retriever. A retriever is trained by minimizing an empirical estimate of the hard-negative NCE loss (5),

$$\widehat{J}_{\text{HARD}}(\theta) = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{\exp\left(s_\theta(x_i, y_{i,1})\right)}{\sum_{k'=1}^{K} \exp\left(s_\theta(x_i, y_{i,k'})\right)} \tag{11}$$

where $(x_1, y_{1,1}) \dots (x_N, y_{N,1})$ denote N mention-entity pairs in training data, and $y_{i,2} \dots y_{i,K} \sim h(\cdot|x_i, y_{i,1})$ are $K-1$ negative entities for the i-th mention. We vary the choice of negatives as follows.

- Random: The negatives are sampled uniformly at random from all entities in training data.

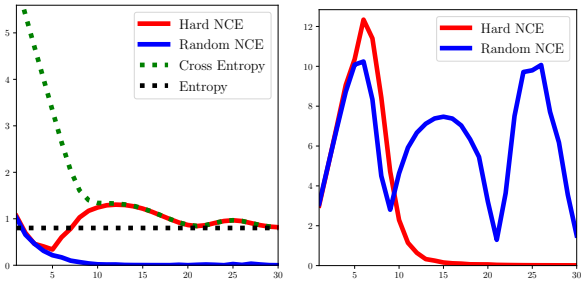

Figure 1: Synthetic experiments. We use a feedforward network to estimate the population distribution by minimizing sampled cross entropy in each step (x-axis). We show the NCE loss (left) and the norm of the gradient bias (right) using hard vs random negatives.

- Hard: The negatives are sampled from (8) each epoch. That is, in the beginning of each training pass, for each i we sample entities $y_{i,2} \dots y_{i,K}$ from $\mathcal{Y} \setminus \{y_{i,1}\}$ without replacement with probabilities proportional to $\exp\left(s_\theta(x_i, y_{i,k})\right)$. This is slightly different from, and simpler than, the original hard negative mining strategy of Gillick et al. (2019) which pretrains the model using random negatives then greedily adds negative entities that score higher than the gold.
- Mixed-p: p percent of the negatives are hard, the rest are random. Previous works have shown that such a combination of random and hard negatives can be effective. We find the performance is not sensitive to the value of p (Appendix A).

We experimented with in-batch sampling as done in previous works (e.g., Gillick et al. (2019)), but found sampling from all training data to be as effective and more straightforward (e.g., the number of random negatives is explicitly unrelated to the batch size). We use $K = 64$ in all experiments.

Training a reranker. We use JOINT only for reranking by minimizing (11) with top-63 negatives given by a fixed retriever, where we vary the choice of retriever. We also investigate other architectures for reranking such as the poly-encoder and the sum-of-max encoder, but we find the full cross attention of JOINT to be indispensable. Details of reranking experiments can be found in Appendix B.

Other details. All models are trained up to 4 epochs using Adam. We tune the learning rate over $\{5\text{e}{-5}, 2\text{e}{-5}, 1\text{e}{-5}\}$ on validation data. We use the training batch size of 4 mentions for all models except for JOINT, for which we use 2. Training time is roughly half a day on a single NVIDIA A100 GPU for all models, except the SOM retriever which takes 1-2 days.

Model	Negatives	Val	Test
BM25	–	76.22	69.13
Wu et al. (2020)	Mixed (10 hard)	91.44	82.06
DUAL	Random	91.08	81.80
	Hard	91.99	84.87
	Mixed-50	91.75	84.16
DUAL-(10)	Hard	91.57	83.08
POLY-16	Random	91.05	81.73
	Hard	92.08	84.07
	Mixed-50	92.18	84.34
MULTI-8	Random	91.13	82.44
	Hard	92.35	84.94
	Mixed-50	92.76	84.11
SOM	Random	92.51	87.62
	Hard	94.49	88.68
	Mixed-50	**94.66**	**89.62**

Table 1: Top-64 recalls over different choices of architecture and negative examples for a retriever trained by NCE. Wu et al. (2020) train a dual encoder by NCE with 10 hard negatives. DUAL-(10) is DUAL trained with the score-adjusted loss (10).

6.4 Bias

We conduct experiments on synthetic data to empirically validate our bias analysis in Section 3.1. Analogous experiments on Zeshel with similar findings can be found in Appendix C.

We construct a population distribution over 1000 labels with small entropy to represent the peaky conditional label distribution $\mathbf{pop}(y|x)$. We use a feedforward network with one ReLU layer to estimate this distribution by minimizing the empirical cross-entropy loss based on 128 iid samples per update. At each update, we compute cross-entropy (2) exactly, and estimate NCE (5) with 4 negative samples by Monte Carlo (10 simulations).

Figure 1 plots the value of the loss function (left) and the norm of the gradient bias (right) across updates. We first observe that hard NCE yields an accurate estimate of cross entropy even with 4 samples. In contrast, random NCE quickly converges to zero, reflecting the fact that the model can trivially discriminate between the gold and random labels. We next observe that the bias of the gradient of hard NCE vanishes as the model distribution converges to the population distribution, which supports our analysis that the bias becomes smaller as the model improves. The bias remains nonzero for random NCE.

Model	Retriever	Negatives	Joint Reranker	Unnormalized	
				Val	Test
Logeswaran et al. (2019)	BM25	–	base	–	55.08
Logeswaran et al. (2019)+DAP	BM25	–	base	–	55.88
Wu et al. (2020)	DUAL (base)	Mixed (10 hard)	base	–	61.34
Wu et al. (2020)	DUAL (base)	Mixed (10 hard)	large	–	63.03
Ours	DUAL (base)	Hard	base	69.14	65.42
	DUAL (base)	Hard	large	68.31	65.32
	SOM (base)	Hard	base	69.19	66.67
	SOM (base)	Hard	large	70.08	65.95
	SOM (base)	Mixed-50	base	69.22	65.37
	SOM (base)	Mixed-50	large	**70.28**	**67.14**

Table 2: Unnormalized accuracies with two-stage training. DAP refers to domain adaptive pre-training on source and target domains.

Mention	. . . his temporary usurpation of the Imperial throne by invading and seized control of the Battlespire, the purpose of this being to cripple the capacity of the Imperial College of Battlemages, which presented a threat to Tharn's power as Emperor. Mehrunes Dagon was responsible for the **destruction** of Mournhold at the end of the First Era, and apparently also . . .
Random	1. **Mehrunes Dagon** is one of the seventeen Daedric Princes of Oblivion and the primary antagonist of . . . 2. **Daedric Forces of Destruction** were Mehrunes Dagon's personal army, hailing from his realm of Oblivion, the Deadlands. . . . 3. **Weir Gate** is a device used to travel to Battlespire from Tamriel. During the Invasion of the Battlespire, Mehrunes Dagon's forces . . . 4. **Jagar Tharn** was an Imperial Battlemage and personal adviser to Emperor Uriel Septim VII. Tharn used the Staff of Chaos . . . 5. **House Sotha** was one of the minor Houses of Vvardenfell until its destruction by Mehrunes Dagon in the times of Indoril Nerevar. . . . 6. **Imperial Battlespire** was an academy for training of the Battlemages of the Imperial Legion. The Battlespire was moored in . . .
Hard	1. **Fall of Ald'ruhn** was a battle during the Oblivion Crisis. It is one of the winning battles invading in the name of Mehrunes Dagon . . . 2. **Daedric Forces of Destruction** were Mehrunes Dagon's personal army, hailing from his realm of Oblivion, the Deadlands. . . . 3. **House Sotha** was one of the minor Houses of Vvardenfell until its destruction by Mehrunes Dagon in the times of Indoril Nerevar. . . . ✓ 4. **Sack of Mournhold** was an event that occurred during the First Era. It was caused by the Dunmer witch Turala Skeffington . . . 5. **Mehrunes Dagon of the House of Troubles** is a Tribunal Temple quest, available to the Nerevarine in . . . 6. **Oblivion Crisis**, also known as the Great Anguish to the Altmer or the Time of Gates by Mankar Camoran, was a period of major turmoil . . .

Table 3: A retrieval example with hard negative training on Zeshel. We use a SOM retriever trained with random vs hard negatives (92.51 vs 94.66 in top-64 validation recall). We show a validation mention (**destruction**) whose gold entity is retrieved by the hard-negative model but not by the random-negative model. Top entities are shown for each model (title boldfaced); the correct entity is **Sack of Mournhold** (checkmarked).

6.5 Retrieval

Table 1 shows the top-64 recall (i.e., the percentage of mentions whose gold entity is included in the 64 entities with highest scores under a retriever trained by (5)) as we vary architectures and negative examples. We observe that hard and mixed negative examples always yield sizable improvements over random negatives, for all architectures. Our dual encoder substantially outperforms the previous dual encoder recall by Wu et al. (2020), likely due to better optimization such as global vs in-batch random negatives and the proportion of hard negatives. We also train a dual encoder with the bias-corrected loss (10) and find that this does not improve recall. The poly-encoder and the multi-vector models are comparable to but do not improve over the dual encoder. However, the sum-of-max encoder delivers a decisive improvement, especially with hard negatives, pushing the test recall to above 89%. Based on this finding, we use DUAL and SOM for retrieval in later experiments.

6.6 Results

We show our main results in Table 2. Following Wu et al. (2020), we do two-stage training in which we train a DUAL or SOM retriever with hard-negative NCE and train a JOINT reranker to rerank its top-64 candidates. All our models outperform the previous best accuracy of 63.03% by Wu et al. (2020). In fact, our dual encoder retriever using a BERT-base reranker outperforms the dual encoder retriever using a BERT-large reranker (65.42% vs 63.03%). We obtain a clear improvement by switching the retriever from dual encoder to sum-of-max due to its high recall (Table 1). Using a sum-of-max retriever trained with mixed negatives and a BERT-large reranker gives the best result 67.14%.

6.7 Qualitative Analysis

To better understand practical implications of hard negative mining, we compare a SOM retriever trained on Zeshel with random vs hard negatives (92.51 vs 94.66 in top-64 validation recall). The

Model	Accuracy
BLINK without finetuning	80.27
BLINK with finetuning	81.54
DUAL with $p = 0$	82.40
DUAL with $p = 50$	88.01
MULTI-2 with $p = 50$	88.39
MULTI-3 with $p = 50$	87.94

Table 4: Test accuracies on AIDA CoNLL-YAGO. BLINK refers to the two-stage model of Wu et al. (2020) pretrained on Wikipedia. All our models are initialized from the BLINK dual encoder and finetuned using all 5.9 million Wikipedia entities as candidates.

mention categories most frequently improved are Low Overlap (174 mentions) and Multiple Categories (76 mentions) (see Logeswaran et al. (2019) for the definition of these categories), indicating that hard negative mining makes the model less reliant on string matching. A typical example of improvement is shown in Table 3. The random-negative model retrieves person, device, or institution entities because they have more string overlap (e.g. "Mehrunes Dagon", "Battlespire", and "Tharn"). In contrast, the hard-negative model appears to better understand that the mention is referring to a chaotic event like the Fall of Ald'ruhn, Sack of Mournhold, and Oblivion Crisis and rely less on string matching. We hypothesize that this happens because string matching is sufficient to make a correct prediction during training if negative examples are random, but insufficient when they are hard.

To examine the effect of encoder architecture, we also compare a DUAL vs SOM retriever both trained with mixed negatives (91.75 vs 94.66 in top-64 validation recall). The mention categories most frequently improved are again Low Overlap (335 mentions) and Multiple Categories (41 mentions). This indicates that cross attention likewise helps the model less dependent on simple string matching, presumably by allowing for a more expressive class of score functions.

6.8 Results on AIDA

We complement our results on Zeshel with additional experiments on AIDA. We use BLINK, a Wikipedia-pretrained two-stage model (a dual encoder retriever pipelined with a joint reranker, both based on BERT) made available by Wu et al. (2020).[3] We extract the dual encoder module from BLINK and finetune it on AIDA using the training portion. During finetuning, we use all 5.9 million Wikipedia entities as candidates to be consistent with prior work. Because of the large scale of the knowledge base we do not consider SOM and focus on the MULTI-m retriever (DUAL is a special case with $m = 1$). At test time, all models consider all Wikipedia entities as candidates. For both AIDA and the Wikipedia dump, we use the version prepared by the KILT benchmark (Petroni et al., 2020).

Table 4 shows the results. Since Wu et al. (2020) do not report AIDA results, we take the performance of BLINK without and with finetuning from their GitHub repository and the KILT leaderboard.[4] We obtain substantially higher accuracy by mixed-negative training even without reranking.[5] There is no significant improvement from using $m > 1$ in the multi-vector encoder on this task.

7 Conclusions

Hard negatives can often improve NCE in practice, substantially so for entity linking (Gillick et al., 2019), but are used without justification. We have formalized the role of hard negatives in quantifying the bias of the gradient of the contrastive loss with respect to the gradient of the full cross-entropy loss. By jointly optimizing the choice of hard negatives and architectures, we have obtained new state-of-the-art results on the challenging Zeshel dataset (Logeswaran et al., 2019).

Acknowledgements

This work was supported by the Google Faculty Research Awards Program. We thank Ledell Wu for many clarifications on the BLINK paper.

References

Yoshua Bengio and Jean-Sébastien Senécal. 2008. Adaptive importance sampling to accelerate training of a neural probabilistic language model. IEEE Transactions on Neural Networks, 19(4):713–722.

Guy Blanc and Steffen Rendle. 2018. Adaptive sampled softmax with kernel based sampling. In

[3] https://github.com/facebookresearch/BLINK

[4] https://ai.facebook.com/tools/kilt/ (as of April 8, 2021)

[5] We find that reranking does not improve accuracy on this task, likely because the task does not require as much reading comprehension as Zeshel.

International Conference on Machine Learning, pages 590–599.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In International conference on machine learning, pages 1597–1607. PMLR.

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.

Thibault Févry, Nicholas FitzGerald, Livio Baldini Soares, and Tom Kwiatkowski. 2020. Empirical evaluation of pretraining strategies for supervised entity linking. In Automated Knowledge Base Construction.

Octavian-Eugen Ganea and Thomas Hofmann. 2017. Deep joint entity disambiguation with local neural attention. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 2619–2629, Copenhagen, Denmark. Association for Computational Linguistics.

Daniel Gillick, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldridge, Eugene Ie, and Diego Garcia-Olano. 2019. Learning dense representations for entity retrieval. In Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL), pages 528–537, Hong Kong, China. Association for Computational Linguistics.

Nitish Gupta, Sameer Singh, and Dan Roth. 2017. Entity linking via joint encoding of types, descriptions, and context. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 2681–2690, Copenhagen, Denmark. Association for Computational Linguistics.

Michael U Gutmann and Aapo Hyvärinen. 2012. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. The journal of machine learning research, 13(1):307–361.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training. arXiv preprint arXiv:2002.08909.

R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. 2019. Learning deep representations by mutual information estimation and maximization. In International Conference on Learning Representations.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust disambiguation of named entities in text. In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pages 782–792.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In Proceedings of the 22nd ACM international conference on Information & Knowledge Management, pages 2333–2338.

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2020. Poly-encoders: Architectures and pre-training strategies for fast and accurate multi-sentence scoring. In International Conference on Learning Representations.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. arXiv preprint arXiv:2004.04906.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over BERT. In Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020, pages 39–48. ACM.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 6086–6096.

Lajanugen Logeswaran, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, Jacob Devlin, and Honglak Lee. 2019. Zero-shot entity linking by reading entity descriptions. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 3449–3460.

Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2020. Sparse, dense, and attentional representations for text retrieval. arXiv preprint arXiv:2005.00181.

Zhuang Ma and Michael Collins. 2018. Noise contrastive estimation and negative sampling for conditional models: Consistency and statistical efficiency. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 3698–3707, Brussels, Belgium. Association for Computational Linguistics.

Andriy Mnih and Yee Whye Teh. 2012. A fast and simple algorithm for training neural probabilistic language models. In Proceedings of the 29th International Coference on International Conference on Machine Learning, pages 419–426.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vassilis Plachouras, Tim Rocktäschel, et al. 2020. Kilt: a benchmark for knowledge intensive language tasks. arXiv preprint arXiv:2009.02252.

Ankit Singh Rawat, Jiecao Chen, Felix Xinnan X Yu, Ananda Theertha Suresh, and Sanjiv Kumar. 2019. Sampled softmax with random fourier features. In Advances in Neural Information Processing Systems, pages 13857–13867.

Joshua David Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. 2021. Contrastive learning with hard negative samples. In International Conference on Learning Representations.

Karl Stratos. 2019. Noise contrastive estimation. http://karlstratos.com/notes/nce.pdf. Unpublished technical note. Accessed: April 8, 2021.

Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. 2020. What makes for good views for contrastive learning. arXiv preprint arXiv:2005.10243.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2020. Scalable zero-shot entity linking with dense entity retrieval. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 6397–6407.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In International Conference on Learning Representations.

A Percentage of Hard Negatives

We show top-64 validation recalls with varying values of the hard negative percentage p in training below:

Mixed-p (%)	DUAL	MULTI-8	SOM
0 (Random)	91.08	91.13	92.51
25	92.18	92.74	94.13
50	91.75	92.76	94.66
75	92.24	93.41	94.37
100 (Hard)	92.05	93.27	94.54

The presence of hard negatives is clearly helpful, but the exact choice of $p > 0$ is not as important. We choose $p = 50$ because we find that the presence of some random negatives often gives slight yet consistent improvement.

B Reranking Experiments

We show the normalized and unnormalized accuracy of a reranker as we change the architecture while holding the retriever fixed:

Model	Normalized		Unnormalized	
	Val	Test	Val	Test
DUAL	60.43	62.49	54.87	54.73
POLY-16	60.37	60.98	54.82	53.37
POLY-64	60.80	61.88	55.20	54.15
POLY-128	60.60	62.72	55.03	54.92
MULTI-8	61.56	62.65	55.90	54.87
MULTI-64	61.94	62.94	56.23	55.15
MULTI-128	61.67	62.95	55.98	55.17
SOM	65.38	65.24	59.35	57.04
GENPOLY-128	65.89	64.98	59.82	56.82
JOINT	**76.17**	**74.90**	**69.14**	**65.42**
Logeswaran et al.	76.06	75.06	–	55.08
Wu et al.	78.24	76.58	–	–
JOINT (ours)	78.82	77.09	58.77	56.56

GENPOLY-m denotes a generalized version of the poly-encoder in which we use m leftmost entity embeddings rather than one: $s_\theta(x, y) = 1_m^\top F_{1:m}(y)^\top C_m(x, y)$. We use a trained dual encoder with 91.93% and 83.48% validation/test recalls as a fixed retriever. The accuracy increases with the complexity of the reranker. The dual encoder and the poly-encoder are comparable, but the multi-vector, the sum-of-max, and the generalized poly-encoder achieve substantially higher accuracies. Not surprisingly, the joint encoder achieves the best performance. We additionally show reranking results using the BM25 candidates provided in the Zeshel dataset for comparison with existing results. Our implementation of JOINT with BERT-base obtains comparable accuracies.

C Bias Experiments on Zeshel

We consider the dual encoder $s_\theta(x, y) = E_1(x)^\top F_1(y)$ where E and F are parameterized by BERT-bases. We randomly sample 64 mentions, yielding a total of 128 entities: 64 referenced by the mentions, and 64 whose descriptions contain these mentions. We consider these 128 entities to constitute the entirety of the label space $\mathcal{Y}$. On the 64 mentions, we estimate $J_{\text{CE}}(\theta)$ by normalizing over the 128 entities; we estimate $J_{\text{HARD}}(\theta)$ by normalizing over $K = 8$ candidates where 7 are drawn from a negative distribution: either random, hard, or mixed. Instead of a single-sample estimate as in (11), we draw negative examples 500 times and average the result. We estimate the bias $b(\theta) \in \mathbb{R}^d$ by taking a difference between these two estimates and report the norm below:

Negatives	$\|b(\theta_{\text{CE}})\|$	$\|b(\theta_{\text{RAND}})\|$
Random	16.33	166.38
Hard	0.68	0.09
Mixed-50	1.20	0.90

We consider two parameter locations. θ_{CE} is obtained by minimizing the cross-entropy loss (92.19% accuracy). θ_{RAND} is obtained by NCE with random negatives (60% accuracy). The bias is drastically smaller when negative examples are drawn from the model instead of randomly. Mixed negatives yield comparably small biases. With random negatives, the bias is much larger at θ_{RAND} since $\nabla J_{\text{CE}}(\theta_{\text{RAND}})$ is large. In contrast, hard and mixed negatives again yield small biases.