ReProCon: Scalable and Resource-Efficient Few-Shot Biomedical Named Entity Recognition

Anonymous ACL submission

Abstract

Named Entity Recognition (NER) in biomedical domains faces challenges from data scarcity and imbalanced label distributions with fine-grained entity types. We propose ReProCon, a few-shot NER framework combining multi-prototype modeling, cosine-contrastive learning, and Reptile meta-learning. By representing each category with multiple prototypes, ReProCon captures semantic variability while ensuring strong interclass separation through cosine-contrastive objectives. Using a lightweight fastText + BiLSTM encoder, ReProCon achieves macro-F_1 scores comparable to BERT-based baselines ($\sim$99% performance) with significantly lower memory usage. The model remains stable with 30% label budget and drops only 7.8% in F_1 when expanding from 19 to 50 categories, outperforming SpanProto and CONTaiNER which degrade 10-32% on Few-NERD. Ablation studies confirm the importance of multi-prototype modeling and contrastive learning for managing class imbalance. Despite challenges with label ambiguity, ReProCon demonstrates state-of-the-art performance in resource-limited settings, making it suitable for biomedical applications where annotation is expensive and computational resources are constrained.

1 Introduction

Named Entity Recognition (NER) identifies and categorizes entity mentions in unstructured text, serving as a foundation for downstream biomedical NLP tasks, including knowledge graph construction, literature curation, and clinical decision support. In biomedical domains, NER is particularly challenging due to the high cost of annotation and the long-tailed label distributions in public corpora, such as MedMentions (Mohan and Li, 2019), BC2GM (Krallinger et al., 2008), and NCBI-Disease (Doğan et al., 2014), which often contain contextually variable entity categories.

Few-shot learning mitigates the data bottleneck by training on only a handful of labeled examples. Although recent work shows promise on general domain benchmarks (e.g., Few-NERD (Ding et al., 2021)), NER has achieved limited success in the biomedical domain with a large number of entity types. In such cases, existing approaches often suffer from poor performance due to the overwhelming diversity of categories and a lack of ontological awareness.

To address these challenges, we propose **ReProCon**, a prototype-based cosine-contrastive NER framework that addresses these gaps. The key components are the following:

1. **Multi-prototype modeling**: Each category has M distinct prototypes that capture various contexts of biomedical entities, designed to be angularly separated for better differentiation.

2. **Cosine supervised-contrastive objective**: The optimization aims for the entity spans to align closely with their corresponding category prototypes while being angularly distanced from others.

3. **First-order meta-updates (Reptile)**: Episode-based updates enable rapid parameter adaptation without the computational burden of second-order gradients (Nichol et al., 2018).

Empirical evaluation in MedMentions demonstrates that ReProCon effectively tackles imbalanced samples and hierarchical complexity of the dataset, such as concept structures based on the Unified Medical Language System (UMLS) (Bodenreider, 2004), while achieving performance comparable to or exceeding state-of-the-art methods, such as BioBERT (Lee et al., 2020) and SpanProto (Wang et al., 2022), with improved computational efficiency.

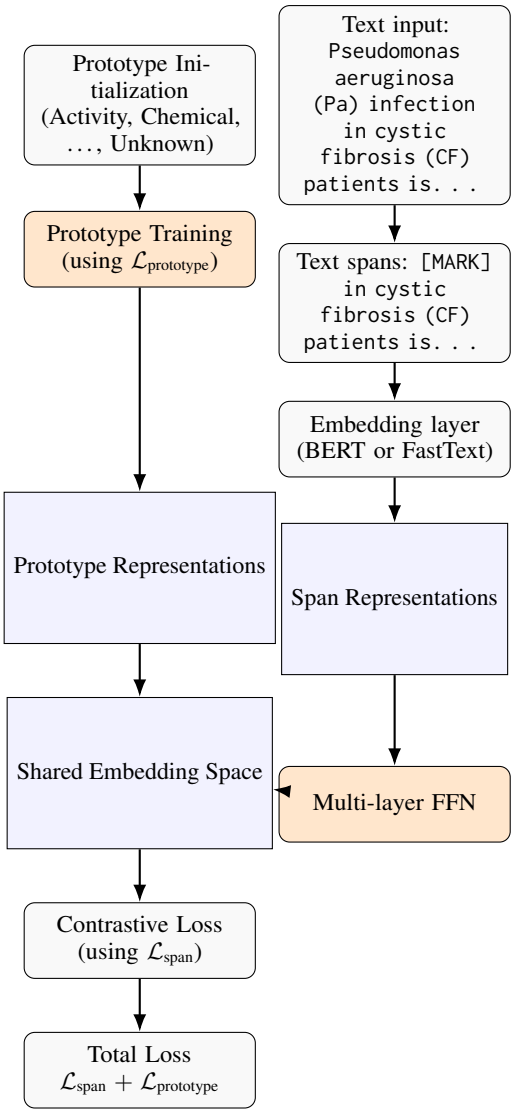

Figure 1: A high-level overview of the model pipeline for a single task, where orange boxes indicate learnable parameters. The system begins with the initialization and training of the prototype. Text inputs are tokenized into spans, embedded (e.g., via BERT (Devlin et al., 2019)), and transformed into span representations.

2 Related Work

Biomedical PLMs. Models like BioBERT, PubMedBERT (Gu et al., 2021), and SciBERT (Beltagy et al., 2019) utilize domain-specific pre-training on PubMed abstracts, achieving substantial improvements in biomedical NER. These results underscore the vocabulary and context mismatch between general and biomedical domains.

Prototypical Networks. Classic prototypical networks assume one centroid per category (Snell et al., 2017). ReProCon extends this by incorporating multiple prototypes per category, enabling finer-grained representation of biomedical entities. We adopt the hyperspherical prototype approach (Mettes et al., 2019) with a repulsion loss to maximize prototype separation.

Meta-Learning. While MAML (Finn et al., 2017) uses second-order gradients for task adaptation, Reptile provides a simpler first-order alternative that eliminates computational overhead while retaining strong few-shot performance. We adopt Reptile for its efficiency and stability in handling noisy biomedical data.

Supervised Contrastive Learning. Recent work (Khosla et al., 2021; Das et al., 2022) applies contrastive learning to NER tasks. We extend this with multi-prototype contrastive loss using category-wise minimum pooling, enabling both semantic flexibility and interclass separation crucial for fine-grained biomedical classification.

3 Data Processing

We normalize whitespace and remove text in parentheses/brackets that often duplicates entities. The MedMentions corpus uses UMLS semantic type IDs with highly imbalanced distributions and multi-label ambiguity. To address this, we apply depth constraints (levels 3 or 4) and low-frequency merging (thresholds: 100 for level 3, 50 for level 4) to reduce sparsity, yielding 19- and 50-way classification settings. For multi-label entities, we use PageRank centrality on a co-occurrence graph to select the most representative type (Brin and Page, 1998). We generate text spans up to 8 tokens using SpaCy (Honnibal et al., 2020), marking positions with `[MARK_POSITION]` to disambiguate multiple instances.

3.1 Task Sampling Strategy for Reptile Meta-Learning

Let K denote the number of support shots and define the split ratio of category-wise training as $r_{\text{train}} \in \{0.3, 0.4, \ldots, 0.8\}$. [1] For each episode, we:

1. sample N (either 19 or 50 in this study) categories uniformly,
2. draw K support examples per category from the training pool,
3. draw validation and query examples from their respective pools.

[1] Validation and query splits are fixed at 10% and $(1 - r_{\text{train}}) - 0.10$, respectively.

To control memory usage, we limit pool sizes to 30,000 (support), 500 (validation), and 400 (query) instances per category and pregenerate 200 episodic task sets. The learning process performs inner loop updates followed by a meta-update with step size α.

4 Model Training

4.1 Notation

Let N be the number of semantic categories, M the number of prototypes per category, and $D = 50$ the dimensionality of each prototype. The shared span-representation space has dimensionality $d_{\text{representation}} = 512$. For the fastText branch, we use static embeddings of size $d_{\text{embed}} = 300$ and sinusoidal positional encodings of size $d_{\text{position}} = 200$. We initialized all weights with Xavier Uniform (Glorot and Bengio, 2010) under a fixed seed 42 for reproducibility. We ran all experiments on a single NVIDIA L4 GPU (22.5GB VRAM) within Google Colab's high-RAM environment.[2]

4.2 Prototype Matrix

Each category maintains $M = 10$ learnable prototypes to capture semantic diversity. The resulting tensor is

$$\mathbf{P} \in \mathbb{R}^{(N \times M) \times D}.$$

Every prototype vector is ℓ_2 normalized after each meta-update to calculate cosine similarity.

4.3 Context Encoder (fastText Setting)

Given a token sequence of length $T = 300$, we concatenate static fastText embeddings(Bojanowski et al., 2017) with positional encodings, described as

$$\mathbf{X} \in \mathbb{R}^{T \times (d_{\text{embed}} + d_{\text{position}})}.$$

A single-layer bidirectional LSTM(Graves et al., 2005) with hidden size $h = 1024$ per direction produces contextual states:

$$\mathbf{H} = \text{BiLSTM}(\mathbf{X}) \in \mathbb{R}^{T \times 2h}.$$

A linear projection $\mathbf{W}_p \in \mathbb{R}^{2h \times d_{\text{representation}}}$ projects token representations in the $d_{\text{representation}}$ space, to align with the output of the BERT encoder in our alternative configuration.

[2]Code and data will be made available upon acceptance.

4.4 Sinusoidal Positional Encoding

We adopt the sinusoidal encoding from (Vaswani et al., 2017). For positions $p \in [0, T-1]$ and dimensions $i \in \left[0, \frac{d_{\text{pos}}}{2} - 1\right]$

$$P_{p,\,2i} = \sin\!\left(p/10000^{2i/d_{\text{position}}}\right), \quad (1)$$

$$P_{p,\,2i+1} = \cos\!\left(p/10000^{2i/d_{\text{position}}}\right), \quad (2)$$

The matrix $\mathbf{P} \in \mathbb{R}^{T \times d_{\text{position}}}$ is concatenated with $\mathbf{X}$ along the feature dimension.

4.5 Span Projection Network

The span embedding of each candidate $\mathbf{s} \in \mathbb{R}^{d_{\text{representation}}}$ is processed by a feed-forward network of L layers:

$$\mathbf{v}^{(\ell+1)} = \text{Dropout}\Big(\text{GELU}\Big(\text{BatchNorm}\big(\mathbf{W}^{(\ell)}\mathbf{v}^{(\ell)} + \mathbf{b}^{(\ell)}\big)\Big)\Big), \mathbf{v}^{(0)} = \mathbf{s}.$$

A final affine transformation followed by Layer Normalization (Ba et al., 2016) projects into the prototype space:

$$\mathbf{z} = \text{LayerNorm}\big(\mathbf{W}^{(\text{out})}\mathbf{v}^{(L)} + \mathbf{b}^{(\text{out})}\big) \in \mathbb{R}^{D}.$$

We computed cosine similarity between the $\mathbf{z}$ and prototype vectors during training and inference. In this study, we set $L = 1$ to avoid overfitting.

4.6 Span Representation Embedding

We evaluate two span-encoding strategies:

1. **fastText + BiLSTM branch** For each SpaCy token, we obtain a fastText vector $\mathbf{e} \in \mathbb{R}^{d_{\text{embed}}}$ and apply the weighted mean for multi-word tokens:

$$\mathbf{e} = \sum_{i=0}^{C-1} w_i\,\hat{\mathbf{e}}_i, \quad w_i = 1 + 0.1\left(\tfrac{C-i}{2}\right)^2, \quad \hat{\mathbf{e}}_i = \frac{\mathbf{e}_i}{\|\mathbf{e}_i\|_2}.$$

   The token sequence is right-padded to length T with zeros, and concatenated with positional encodings:

$$\mathbf{X} = \big[\mathbf{E} \parallel \mathbf{P}\big] \in \mathbb{R}^{T \times (d_{\text{embed}} + d_{\text{position}})}.$$

   A single layer BiLSTM (h=1024 per direction) produces contextual states $\mathbf{H} \in \mathbb{R}^{T \times 2h}$. The hidden state at the marker position becomes the span representation $\mathbf{s} \in \mathbb{R}^{d_{\text{representation}}}$.

2. **BERT branch** Let $(s_{\text{start}}, s_{\text{end}})$ denote the range of subtoken indexes of the marked span (computed by aligning SpaCy tokens with the BERT wordpiece sequence and skipping all [CLS]/[SEP] markers). We feed the entire sequence into the vanilla BERT-base-cased model and obtain hidden states $\mathbf{H} \in \mathbb{R}^{T \times d_{\text{bert}}}$. The span representation $\mathbf{s} \in \mathbb{R}^{d_{\text{representation}}}$ is a pooled tensor, using the mean-max scheme. As BERT inherently captures bidirectional context, this branch does not need an extra BiLSTM layer.

4.7 Loss Functions

The total objective is the sum of a *prototype repulsion loss* term and a *span alignment loss* term:

$$\mathcal{L} = \mathcal{L}_{\text{proto}} + \mathcal{L}_{\text{span}}.$$

Prototype repulsion loss. Inspired by a methodology suggested in EPNet, we maximize the angular separation between $N \times M$ prototypes $\{\mathbf{p}_0, \ldots, \mathbf{p}_{N \times M - 1}\}$:

$$\mathcal{L}_{\text{proto}} = \frac{1}{NM} \sum_{i=0}^{NM-1} \max_{j \neq i} \big(\cos(\mathbf{p}_i, \mathbf{p}_j) + 1\big).$$

The offset $+1$ ensures a positive quantity and penalizes highly aligned pairs.

Span alignment loss. Let $\{\mathbf{q}_0, \ldots, \mathbf{q}_{NK-1}\}$ be the projected span vectors of a training episode (K spans per category). We first compute the squared angular distance $H_{ij} = (1 - \cos(\mathbf{p}_i, \mathbf{q}_j))^2$. For each category c, we take the minimum distance between its M prototypes:

$$\widetilde{H}_{c,j} = \min_{cM \leq k < (c+1)M} H_{kj}.$$

The span alignment loss encourages each query span to align closely with the prototype of its category while remaining distant from the others, enabling a robust classification of biomedical entities:

$$\mathcal{L}_{\text{span}} = \sum_{c=0}^{N-1} \frac{\frac{1}{K} \sum_{j=cK}^{(c+1)K-1} \widetilde{H}_{c,j}}{\sum_{j=0}^{NK-1} \widetilde{H}_{c,j}}.$$

4.8 Meta-Learning with Reptile

Each meta-episode involves (1) sampling a few-shot task from the pool, (2) fine-tuning a task model in its support set under the contrastive objective of Section 4.7, and (3) performing a first-order meta-learning update to the meta model. Algorithm 1 describes the procedure.

We set inner epochs to $E = 5$ (fastText) and $E = 3$ (BERT), with learning rates $\gamma = 5 \times 10^{-4}$ and $\gamma = 10^{-3}$, respectively, optimized by cross-validation. The meta-step sizes are $\alpha = 0.4$ and $\alpha = 0.5$. Cosine decay scheduling (Loshchilov and Hutter, 2017) and gradient clipping (Pascanu et al., 2013) (max norm 1.0) stabilize the training.

Algorithm 1 Reptile Meta-Training Loop

Require: Initial meta-parameters θ_0, pool of episodic tasks $\mathcal{T}$, validation set $\mathcal{V}$, meta step size α, task learning rate γ, inner epochs $\mathcal{E}$, outer epochs M, patience $\mathcal{P}$

Ensure: Best meta-parameters θ^*

```
θ ← θ_0
for outer epoch m = 1 to M do
    Sample few-shot task S ~ T
    clone meta-model: θ'_0 ← θ
    Initialize task optimizer (learning rate = γ) and cosine scheduler
    for inner epoch ⌉ = 1 to E do
        Generate span projections
        Compute L_proto, L_span
        L ← L_proto + L_span
        Perform a gradient descent step on θ'
    end for
    θ'_1 ← θ'                    ▷ fine-tuned parameters
    Reptile update:
        θ ← θ + α(θ'_1 − θ'_0)        (Reptile)
    ℓ2-normalize all prototype vectors in θ
    Evaluate F1 on V; keep best weights θ*
    if no F1 improvement for P outer epochs then
        break                     ▷ early stopping
    end if
end for
return θ*
```

4.9 Hard-Negative Training

After the initial meta-training phase, we identify hard-negative spans that the model assigns high scores to despite belonging to other categories. We conducted a second meta-training round using the resampled support sets under the same hyperparameters. We compared the performance of the model against a model trained without hard-negative mining to evaluate changes in classification accuracy.

5 Discussion

5.1 Prototype Representation and Optimization

We adopt a multi-prototype strategy, allocating M distinct prototypical vectors per category to capture the variability within the same category. We chose cosine similarity over Euclidean distance for two reasons:

1. In high-dimensional spaces, Euclidean distances are less effective metrics (Aggarwal et al., 2001), whereas the cosine distance remains discriminative under ℓ_2 normalization (Korenius et al., 2007).

2. Our projection vectors are already normalized, so angular objectives converge more smoothly and yield more precise decision boundaries.

UMAP visualization of the vectors projected from the query set trained with a 0.3 training split ratio of the NER model based on fastText of 19 ways via the UMAP dimensionality reduction (McInnes et al., 2020) indicates that some categories have converged into discrete clusters; however, the number of clusters is less than the number of prototypes assigned per category that we set.

5.2 Performance Comparison

We benchmarked two embedding backbones: a lightweight fastText + BiLSTM encoder and a contextualized BERT model. As reported in Table 1, fastText embedding achieved comparable macro-F_1 performance compared to BERT embedding. Considering its minimal memory footprint, faster inference, and straightforward deployment, the fastText pipeline may be a cost-effective option when working with constrained data and computing resources.

We further assessed robustness under varying data budgets, training in 30% -80% of available samples. In particular, F_1 remains stable or improves with fewer examples (Table 1), underscoring the data efficiency of our reptile-based few-shot paradigm.

Analysis of the confusion matrix for one of our models, which uses a token embedding based on fastText + BiLSTM, trained on 30% of the total samples, reveals patterns of misclassification that can inform future improvements. The model demonstrates reasonable accuracy, with strong performance in specific categories such as `Group` (352 correct predictions), `Injury or Poisoning` (346), and `Organization` (326), where the recall exceeds 80%. These categories appear to have more distinct features, possibly due to their semantic uniqueness in the biomedical domain.

Table 1: Macro-F_1 (%) of **fastText** vs. **BERT** under various training-query split ratios for 19-way and 50-way classification.

19-way	0.8	0.7	0.6	0.5	0.4	0.3
fastText	43.28	46.39	49.45	47.58	50.18	50.80
BERT	46.37	45.97	48.26	47.68	48.09	51.03
Mean	fastText = 47.95 | BERT = 47.90					
50-way	0.8	0.7	0.6	0.5	0.4	0.3
fastText	46.83	50.50	49.89	51.89	50.71	46.85
BERT	53.79	54.60	53.78	52.03	51.11	46.73
Mean	fastText = 49.45 | BERT = 52.01					

Notable confusion patterns include misclassifications between `Anatomical Structure` and `Substance` (35 and 55 instances, respectively), likely due to shared lexical features in biological texts (e.g., tissue or compound). `Phenomenon or Process` and `Natural Phenomenon or Process` show reciprocal errors (37 each), indicating potential label ambiguity in the training data or insufficient examples to learn fine-grained distinctions. The `Activity` is confused with `Occupational Activity` (45) and `Phenomenon or Process` (54), underscoring the model's difficulty in distinguishing between action-oriented and procedural entities, a challenge exacerbated by limited shots in our few-shot setup.

5.3 Class Scalability Analysis

Our model loses only **7.8%** when the label space expands from 19 to 50 biomedical types, while representative baselines (NNShot, StructShot, CONTaiNER) show 10 to 32% degradation when categories double in Few-NERD. We attribute this robustness to our supervised contrastive objective. By minimizing category-wise minimum distances between spans and multiple prototypes, gradient domination from primary categories is alleviated, yielding balanced decision boundaries even under extreme category imbalance.

Table 2: Ablation on 19-way task (Macro-F_1, %).

Configuration	Change	Macro-F_1
Complete model	—	**50.80**
Single prototype	$M = 1$	49.13
Hard-neg.	disabling hard-neg. mining	56.66
CE loss	no contrastive	2.70

5.4 Ablation Study

To isolate the contribution of each architectural decision, we ablate three key components while keeping all other hyperparameters fixed (fastText + BiLSTM backbone, 0.3 split). Table 2 investigates the contribution of each architectural or training ingredient to macro-F_1 in general.

Replacing the supervised-contrastive objective with vanilla cross entropy causes a collapse to near-random performance (2.7%). It highlights that the contrastive learning process makes a substantial contribution in a few-shot setting. This property is crucial when there is an extreme imbalance of samples in each category.

Collapsing each category with a single centroid reduces F_1 by approximately 1.7 percentage points. The result confirms our intuition that biomedical entity categories are semantically diverse (e.g., gene symbols vs. full protein names) and therefore benefit from distributed rather than monolithic representations.

Turning off hard-negative sampling and returning to uniformly selected negatives increased performance to 56.66%. We hypothesize that gradient instability causes this, which is exacerbated by hard-negative mining a handful of high-loss examples, producing steep, noisy gradients that counteract the regularizing effect of contrastive loss. We believe that when the training set is already small, this variance outweighs the potential benefit of focusing on hard-negative samples.

5.5 Tag Set Extension

We experimented with adding previously unseen labels during training to assess the model's ability to retain knowledge while integrating new information. From the 19-way label set, we randomly split labels into groups and trained the model first on two-thirds of labels (Phase 1), then on the complete set (Phase 2). Performance on the base subset of labels remained stable or improved despite adding unseen labels, with Split A showing statistically significant improvement ($p = 0.039$). New labels can be added late in training without performance degradation, demonstrating the model's robustness to changing label distributions.

6 Conclusion

We proposed ReProCon, a cosine-contrastive few-shot NER framework for biomedical text that integrates supervised contrastive learning with Reptile meta-updates. Using a lightweight **fastText + BiLSTM** encoder, ReProCon achieves ∼99% of a baseline Macro-F_1 based on BERT while requiring significantly less memory. Performance remains stable with a label budget of 30% and decreases by only 3.95 percentage points (7.8%) when expanding from 19 to 50 categories. These results demonstrate that state-of-the-art biomedical few-shot NER is achievable without large language models, making ReProCon well-suited for resource-constrained environments.

In future work, we plan to:

1. Conduct cross-domain meta-training from PubMed abstracts to clinical notes and patent texts to evaluate domain-shift robustness.

2. Comparison of performances between models from other studies, by training on biomedical datasets.

7 Limitations

While ReProCon demonstrates strong performance in biomedical few-shot NER, several limitations should be noted. First, the model struggles with label ambiguity when entities can belong to multiple overlapping categories. Second, performance evaluation is limited to the MedMentions dataset; cross-domain generalization to clinical notes or patent texts requires further validation. Third, the current prototype allocation strategy uses a fixed number of prototypes per category ($M = 10$), which may not be optimal for all semantic types. Finally, hard-negative mining showed performance degradation in our experiments, suggesting that alternative sampling strategies may be needed for very small training sets.

8 Ethics Statement

This work focuses on biomedical named entity recognition, which has important applications in healthcare and medical research. The MedMentions dataset used in this study is publicly available and deidentified. We acknowledge that NER systems applied to clinical data must be carefully

validated and should not be used for clinical decision making without appropriate oversight and regulatory approval. Future deployments should consider potential biases in training data and ensure compliance with healthcare privacy regulations.

References

Charu C. Aggarwal, Alexander Hinneburg, and Daniel A. Keim. 2001. On the surprising behavior of distance metrics in high dimensional space. In *Database Theory — ICDT 2001*, pages 420–434, Berlin, Heidelberg. Springer Berlin Heidelberg.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. *Preprint*, arXiv:1607.06450.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. *Preprint*, arXiv:1903.10676.

Olivier Bodenreider. 2004. The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic Acids Research*, 32:D267–D270.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117.

Sarkar Snigdha Sarathi Das, Arzoo Katiyar, Rebecca Passonneau, and Rui Zhang. 2022. CONTaiNER: Few-shot named entity recognition via contrastive learning. *Preprint*, arXiv:2109.07589.

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.

Ning Ding, Guangwei Xu, Yulin Chen, Xiaobin Wang, Xu Han, Pengjun Xie, Haitao Zheng, and Zhiyuan Liu. 2021. Few-NERD: A few-shot named entity recognition dataset. *Preprint*, arXiv:2105.07464.

Rezarta Islamaj Doğan, Robert Leaman, and Zhiyong Lu. 2014. NCBI disease corpus: a resource for disease name recognition and concept normalization. *Journal of Biomedical Informatics*, 47:1–10.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. *Preprint*, arXiv:1703.03400.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9, pages 249–256, Chia Laguna Resort, Sardinia, Italy. PMLR.

Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. 2005. Bidirectional LSTM networks for improved phoneme classification and recognition. In *Artificial Neural Networks: Formal Models and Their Applications – ICANN 2005*, pages 799–804, Berlin, Heidelberg. Springer Berlin Heidelberg.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare*, 3(1):1–23.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength natural language processing in Python.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2021. Supervised contrastive learning. *Preprint*, arXiv:2004.11362.

Tuomo Korenius, Jorma Laurikkala, and Martti Juhola. 2007. On principal component analysis, cosine and Euclidean measures in information retrieval. *Information Sciences*, 177(22):4893–4905.

M. Krallinger, A. Morgan, L. Smith, F. Leitner, L. Tanabe, J. Wilbur, L. Hirschman, and A. Valencia. 2008. The BioCreative II—critical assessment for information extraction in biology challenge. *Genome Biology*, 9:S1.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Ilya Loshchilov and Frank Hutter. 2017. SGDR: Stochastic gradient descent with warm restarts. *Preprint*, arXiv:1608.03983.

Leland McInnes, John Healy, and James Melville. 2020. UMAP: Uniform manifold approximation and projection for dimension reduction. *Preprint*, arXiv:1802.03426.

Pascal Mettes, Elise van der Pol, and Cees G. M. Snoek. 2019. Hyperspherical prototype networks. *Preprint*, arXiv:1901.10514.

Sunil Mohan and Donghui Li. 2019. Medmentions: A large biomedical corpus annotated with UMLS concepts. *Preprint*, arXiv:1902.09476.

Alex Nichol, Joshua Achiam, and John Schulman. 2018. On first-order meta-learning algorithms. *Preprint*, arXiv:1803.02999.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28, pages 1310–1318, Atlanta, Georgia, USA. PMLR.

Jake Snell, Kevin Swersky, and Richard S. Zemel. 2017. Prototypical networks for few-shot learning. *Preprint*, arXiv:1703.05175.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Preprint*, arXiv:1706.03762.

Jianing Wang, Chengcheng Han, Chengyu Wang, Chuanqi Tan, Minghui Qiu, Songfang Huang, Jun Huang, and Ming Gao. 2022. SpanProto: A two-stage span-based prototypical network for few-shot named entity recognition. *Preprint*, arXiv:2210.09049.