BondBERT: An ensemble-based model for named entity recognition in materials science texts

Stanford CS 224N Custom Project

Bella Crouch
Institute for Computational and Mathematical Enginering
Stanford University
bcrouch@stanford.edu

Abstract

Existing tranformer-based models have typically shown poor performance when applied to named entity recoginition (NER) tasks in materials science due to the interdisciplinary nature of literature in the field. This project proposes an NER architecture inspired by the diversity of materials science texts. By ensembling several BERT models pretrained on texts from neighboring scientific fields (chemistry, material physics, biomedicine, and electrical engineering), the proposed architecture facilitates transfer learning across several disciplines relevant to materials science while benefitting from the reductions in model variance brought by ensembled methods. Five variations of ensemble architecture are trialed on the Matscholar corpus, a dataset of IOB-tagged materials science journal articles. The ensembled model is found to out-perform the previous benchmark model with a 7.8% improvement in multiclass F1 score on the development set; however, the ensemble may fail to identify entities in cases where context is domain-specific.

1 Key Information to include

- Mentor: Tony Wang
- Sharing project: No

2 Introduction

Materials science – the study of the physical, chemical, and biological properties of materials to inform engineering design – produces large volumes of experimental data, yet, until recent digitization efforts, has lacked the centralization of data present in other scientific fields (Olivetti et al., 2020). In lieu of central databases for sharing experimental results, the primary mode of communication in the field has traditionally been publications in academic journals. As such, named entity recognition (NER) for materials science literature is a key task for advancing informatics in the field. By identifying entity references in past research studies – such as chemical names, crystal structures, or experimental conditions – NER models enable the rapid parsing of materials physics data. This, in turn, can contribute towards advancements in material discovery and synthesis mediated by deep learning techniques.

Traditional frameworks for NER have, however, had limited success when reapplied within the materials science domain. Materials science is an inherently interdisciplinary field; accordingly, writing in materials science literature exhibits syntactic patterns from several related domains such as chemistry, physics, and biology. Previous attempts (Song et al., 2023) at pretraining transformer-based models on materials science literature have resulted in poor performance, in part due to the diversity of texts in the field. Applying models tuned for scientific NER, such as BioBERT and SciBERT, to the materials science domain has similarly yielded weak performance.

In this work, we draw inspiration from the interdisciplinarity of materials science to propose a new framework for NER in materials literature. We present an ensemble model that aggregates the entity predictions generated by several BERT-based sublearner models, each pretrained on texts

from neighboring scientific fields, to inform a metalearner tuned to the materials science NER task. The proposed architecture aims to incorporate learned knowledge from several disciplines relevant to materials science while benefitting from the reductions in model variance brought by ensemble methods. The resulting metalearner, BondBERT, "bonds" the sublearner predictions together much like the atomic bonds described by the materials literature it parses.

3 Related Work

NER for materials science. Though still nascent, NLP for materials science has attracted a considerable increase in interest in recent years. Song et al. (2023) introduce MatSci-NLP, the first formal benchmark for evaluating NLP on materials science texts. Weston et al. (2019) perform NER on a materials science corpus using stacked BiLSTMs with conditional random fields and Word2Vec embeddings. Trewartha et al. (2022) pretrain a BERT model for materials NER, finding that, although materials domain-specific pretraining leads to marginal improvement in F1 score over existing frameworks for general-purpose scientific text processing, the efficacy of pre-training is highly sensitive to the choice of training corpus. The diversity of subject matters and syntactic patterns across sub-fields of materials science means that the balance of *specificity* for particular facets of materials physics and *breadth* of disparate sub-fields represented in a corpus steer the transformer's performance. Our ensemble approach seeks to target this balance: we achieve specificity by way of pre-training each sublearner on a corpus with narrow scientific focus, and diversity through transfer learning across these "domain expert" sublearners in the ensemble.

NER via ensemble learning. Several past efforts have explored the use of ensemble-based methods for NER. Devlin et al. (2019)'s original BERT paper briefly experiments with ensembling seven BERT models each initialized from different pre-training checkpoints. Cakaloglu and Xu (2019) propose an ensemble that constructs multi-resolution embeddings for each word in the original text through a mixture of token representations by individual sublearners, then passes these ensembled embeddings into a metalearner neural network. In both studies, the sublearners are general-purpose BERT models pre-trained on texts broadly representative of typical English language use; in our work, we aim to use ensembling to facilitate transfer learning *between* distinct domains.

Copara et al. (2020) and Lin et al. (2022) implement transfer learning across scientific sub-domains by ensembling transformer-based models each pretrained on texts from different scientific fields. Each sublearner's entity tag prediction for a token in the original text is taken as the tag with maximum softmax probability among all subwords of the token; then, the ensemble's tag prediction is the majority vote of sublearners for that token. The simplicity of the ensembling scheme is necessitated by differences in subword tokenizations across the sublearners, which means that the ensemble cannot leverage the per-subword embedding representations generated by each sublearner. Our work extends this approach by re-aligning sublearner tokenizations such that the metalearner can make use of the information encoded by the hidden layers of each sublearner when making ensemble predictions.

4 Approach

To emulate the interdisciplinary nature of materials literature, we construct an ensemble NER model of BERT-based sublearners each pretrained via masked language modeling on texts sourced from a scientific field adjacent to materials science. In doing so, we adopt the ensemble learning paradigm where the wisdom of a "crowd" of sublearners, each knowledgeable in a different domain, pool their complementary knowledge for the materials NER task. The sublearners of the ensemble, selected for their pretraining corpuses' diversity and relevance to materials science, are listed in Table 1 below.

Sublearner	Domain of pretraining corpus	Reference
MatBERT	Materials science	Trewartha et al. (2022)
BioBERT	Biomedicine	Lee et al. (2019)
SciBERT	General science	Beltagy et al. (2019)
BatteryBERT	Battery physics	Huang and Cole (2022)
Chemical-BERT	Chemistry	Recobo.ai (2023)

Table 1: Sublearners of BondBERT ensemble

We perform Bayesian hyperparameter optimization to select ideal hyperparmeters for each sublearner, then finetune the sublearner on a corpus of materials science texts for a small number of epochs with the AdamW optimizer. The number of finetuning epochs is restricted to limit the variance of each sublearner model; in particular, this restriction of complexity prevents the case where all five sublearners converge towards similar outputs and lose the benefit of pre-training on different domains.

4.1 Embedding Alignment

At prediction time, the input text is tokenized according to the tokenization scheme for each sublearner model, then passed through the finetuned BERT model to generate per-learner token embeddings. Because each sublearner uses a different tokenization, sequences of characters treated as a single token in one sublearner may be partitioned into several subword tokens in another. This means that, since there is no guarantee of cross-correspondence in what subwords are represented by embeddings at identical positions in different sublearner transformers, the per-learner token embeddings cannot be directly aggregated across the ensemble.

We implement an algorithm of our own design, *AlignEmbeddings*, to align embeddings over all learners in the ensemble. The ensemble-wide alignment algorithm extends spaCy's implementation (Explosion Inc, 2023) of Myers' algorithm for aligning spans across different tokenizations (Myers, 1986), augmenting the sublearner embedding outputs such that they have dimension equal to the maximal cardinality of any partition of subwords across the ensemble (see Figure 1).

The algorithm post-processes sublearner outputs by duplicating the hidden layer and logit predictions of each sublearner for as many repetitions as is necessary to adhere to the newly-aligned ensemble tokenization. In doing so, this alignment enables more sophisticated aggregation schemes (likelihood maximization, adaptive boosting, and hidden layer pooling) than the simple majority voting previously reported in the literature for ensembles of several tokenizations. The implementation of the alignment algorithm is detailed in Algorithm A1.

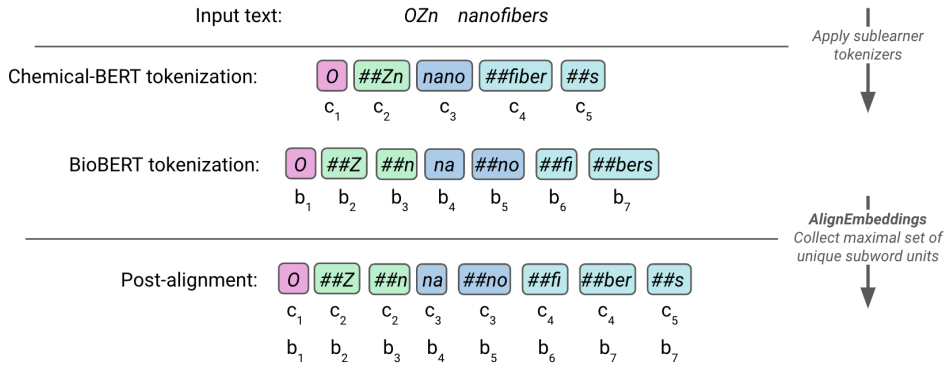

Figure 1: Illustration of *AlignEmbeddings* for generating aligned ensemble tokenizations of input text. Terms of the form c_i and b_j represent subword embeddings for the ith and jth subword of the respective sublearner. Note the duplication of sublearner embeddings post-alignment to match the ensemble-wide tokenization scheme.

4.2 Ensemble Architecture

The aligned sublearners are then ensembled following one of two architectures.

4.2.1 Voting Architecture

In the *voting* architecture, each sublearner independently generates entity tag predictions for the input tokens, which are treated as "votes" for the metalearner prediction. Three voting schemes are implemented.

Majority voting. Each sublearner L votes for the entity tag T with maximum softmax probability, $\arg\max_T p_L(T)$. The metalearner outputs the entity tag with the most votes across all sublearners.

Likelihood Maximization. The joint density of predicted softmax probabilities for tag T across each of ℓ sublearners L is computed as $f(T) = \Pi_{L=1}^{\ell} p_L(T)$. The metalearner outputs the entity tag with maximum likelihood, $\arg\max_T f(T)$

Boosting. Inspired by multiclass Adaboost (Hastie et al., 2009), predictions made by each sublearner are weighted by that sublearner's performance on the training data. This approach is motivated by the pre-training domains of the five sublearners – a sublearner trained on texts from a particular scientific field may be more or less suited at identifying particular entity tags. For example, MatBERT may be more effective at identifying symmetry label tags, a task unique to materials science NLP. Each sublearner L is assigned an error score (1 - per-class precision) for each entity tag class T; this score represents the sublearner's "confidence" in making predictions for that class. Entity predictions are made as follows, where $\hat{Y}_i^L = \arg\max_T p_L(T)_i$ is the entity prediction made by sublearner L on training example i across t possible entity tag choices:

$$\text{err}_L^T = 1 - \text{precision}_L^T$$

$$\alpha_L^T = \log\left(\frac{1 - \text{err}_L^T}{\text{err}_L^T}\right) + \log(t-1)$$

$$\text{Metalearner prediction: } = \arg\max_T \sum_{L=1}^{\ell} \alpha_L^T \mathbb{1}(\hat{Y}_i^L = T)$$

4.2.2 Multi-Learner Embedding Architecture

In the *multi-learner embedding* architecture, the hidden layers of all sublearners are pooled to generate ensemble embeddings for each aligned token. From a linguistic standpoint, we interpret this method as learning token embeddings that incorporate domain-dependent word meaning. As an illustrative example, the word "cell" is used with varied meaning in materials science writing: a biomaterials journal may refer to biological cells, a crystallography paper may describe the unit cell of a crystal, while a third text might reference a electrolytic cell in the context of batteries. The multi-layer embedding architecture draws upon the domain-specific embeddings of each sublearner to aggregate metalearner embeddings able to capture all three meanings.

We extend the work of Cakaloglu and Xu (2019) by applying a mixture model of two cascaded operations, f_{mix} and f_{ensemble}, to produce ensembled representations of tokens post-alignment. For each sublearner L, we generate a per-sublearner embedding for token i as the concatenation of transformer hidden states from the jth hidden layer, $\mathbf{e}_i^{L,j}$, to the kth hidden layer, $\mathbf{e}_i^{L,k}$:

$$\mathbf{x}_i^L = f_{\text{mix}}(\mathbf{e}_i^{L,j}, \mathbf{e}_i^{L,j+1}, \ldots \mathbf{e}_i^{L,k}) =< \mathbf{e}_i^{L,j} \oplus \mathbf{e}_i^{L,j+1}, \oplus \ldots \oplus \mathbf{e}_i^{L,k} >$$

Drawing from the recommendations of the original BERT paper (Devlin et al., 2019), we perform this concatenation over the final four hidden layers of each sublearner (setting $j = 8$, $k = 12$). The resulting ensemble representation is expressed as the set $X_i' = \{\mathbf{x}_i^1, \mathbf{x}_i^2, \ldots, \mathbf{x}_i^\ell\}$ for an ensemble of ℓ sublearners. The aggregated embedding for the token is then:

$$\mathbf{x}_i = f_{\text{ensemble}}(X_i', \mathbf{w})$$

where f_{ensemble} performs a weighted average over the sublearner representations in the set X_i' using learned weights $\mathbf{w}$, with $\sum_{L=1}^{\ell} w_L = 1, w_L \in [0, 1]$.

The metalearner is a neural network that accepts the ensembled embeddings as input and applies further learned weights to generate a predicted probability distribution over the entity tags; the tag with highest predicted probability is then outputted as the metalearner prediction. We implement two metalearner network architectures: a fully-connected feed-forward network of three layers, and a BiLSTM network with two layers. Both networks are trained with cross-entropy loss as the minimization objective.

Schematics of the voting and multi-learner embedding architectures are given in Appendix Figures A1 and A2

5 Experiments

5.1 Data

We develop the ensemble using data sourced from the MatSci-NLP suite of benchmark datasets (Song et al., 2023), the current state-of-the-art natural language benchmark for NLP in the materials domain. We focus our work on the Matscholar NER task outlined by the benchmark, which includes 5,458 training examples of IOB-tagged texts from materials science and battery design. Possible entity tags are {“material name" (MAT), “symmetry label" (SPL), “sample descriptor" (DSC), “material property" (PRO), “material application" (APL), “synthesis method" (SMT), “characterization method" (CMT)}. 1,092 examples (20%) are designated as a held-out test set; the remaining examples are used for model training and validation.

5.2 Evaluation method

In keeping with the MatSci-NLP benchmark, we evaluate model performance by the F1 score for entity predictions, defined as the harmonic mean of precision and recall. Scores are reported over five cross-validation trials on the training data as well as a final score on the held-out test set.

To provide a comparison of performance to the ensemble approach, we reproduce the benchmarks reported by MatSci-NLP using “vanilla", non-ensembled BERT models. As in the MatSci-NLP standard, all benchmark trials are finetuned for 20 epochs with early stopping using the AdamW optimizer, 0.01 weight decay, no warm-up, and a learning rate of 2×10^{-5}.

5.3 Experimental details

To prepare the sublearners for ensembling, 15 iterations of Bayesian optimization for hyperparameter selection are performed over 3 finetuning epochs using Optuna. The five sublearners are then finetuned for 3 epochs using the AdamW optimizer with optimized hyperparameters listed in Table 2 and a dropout rate of 0.1 for all BERT attention layers. Huggingface defaults are used for all other parameters of the sublearners. Finetuning of each sublearner takes approximately 20 minutes on a single NVIDIA T4 GPU.

	MatBERT	BioBERT	Chemical-BERT	BatteryBERT	SciBERT
Learning rate	4.8×10^{-5}	4.4×10^{-5}	4.4×10^{-5}	4.8×10^{-5}	4.5×10^{-5}
AdamW warm-up	369	355	263	86	106
AdamW weight decay	0.0973	0.0210	0.0434	0.0992	0.0926

Table 2: Optimized hyperparameters used for finetuning sublearners before ensembling

5.3.1 Voting Architecture

The five sublearners are then ensembled to implement the voting metalearner. To compute boosting scores α_L^T for the boosted voting scheme, we train the ensemble model for one epoch across all training examples, which runs for 15 minutes on a single NVIDIA T4 GPU.

After running the ensemble of five sublearners, we perform a series of ablation studies that use subsets of only three or four of the five learners. In doing so, we seek to investigate the contribution of each sublearner to the overall ensemble.

5.3.2 Multi-Learner Embedding Architecture

For both the feed-forward and BiLSTM metalearners, ensemble weights $\mathbf{w}$ are initialized using the Xavier uniform initialization, while all other layers are initialized using the PyTorch default Kaiming uniform initialization. The fully-connected network passes ensembled embeddings through three linear layers with ReLU activation; the BiLSTM passes embeddings through a bidirectional LSTM followed by a linear layer with ReLU activation to project BiLSTM outputs into the entity tag dimension. We perform 15 iterations of Bayesian optimization for hyperparameter selection. The metalearner networks are trained for 10 epochs with early-stopping using the optimized hyperparameters given in Table 3.

	Feed-forward network	BiLSTM
Layer dimensions	$\{4 \cdot \dim(\mathbf{x}) \times 512, 512 \times 256, 256 \times t\}$	$\{4 \cdot \dim(\mathbf{x}) \times 512, 512 \times t\}$
Learning rate	2.4×10^{-5}	5.0×10^{-5}
AdamW warm-up	229	228
AdamW weight decay	0.0001	0.033
Dropout rate	0.0551	0.0567

Table 3: Layer specifications and hyperparameters used for metalearner networks in multi-learner embedding architecture. $\dim(\mathbf{x})$ denotes the dimension of token embeddings before concatenation over the final four hidden layers of each sublearner (768); t is the number of IOB entity tags (15)

Feed-forward training runs for approximately 4 hours on an NVIDIA T4 GPU; BiLSTM training runs for approximately 4.5 hours with this same configuration.

5.4 Results

5.4.1 Baselines

To provide a baseline level for performance benchmarking, we report the F1-scores of each *individual* sublearner in Table 4 as the mean over five cross-validation trials, $\pm 2\sigma$. We note that MatBERT, BioBERT, and BatteryBERT have been previously reported to produce state-of-the-art performance on the MatSci-NLP benchmark (Song et al., 2023); our replication study affirms this finding. The bolded best-performing model is taken as the "baseline" performance for all subsequent analysis in this paper.

	MatBERT	BioBERT	Chemical-BERT	BatteryBERT	SciBERT
Validation F1	$0.830_{\pm 0.0015}$	$0.843_{\pm 0.0003}$	$0.753_{\pm 0.0083}$	$\mathbf{0.848}_{\pm 0.0118}$	$0.828_{\pm 0.0010}$

Table 4: Benchmark development set F1 scores of non-ensembled models for NER on Matscholar corpus. Results are reported as the mean across 5 trials $\pm 2\sigma$; best-performing benchmark is bolded.

5.4.2 Ensembling

We report the performance of the ensemble variants in Table 5. For the voting architecture, we additionally perform ablation studies in which all 15 possible subsets of three or four of the five sublearners are independently ensembled. We present voting results using all five sublearners ("full"), as well as the best-performing ablation subset (MatBERT, BioBERT, SciBERT; "ablated") in Table 5. Full results for the ablation study are provided in Appendix Table A1.

Ensemble performance is found to improve significantly after the removal of the Chemical-BERT and BatteryBERT sublearners during ablation. This observation, as well as CUDA memory constraints on available GPUs, motivate the decision to build the multi-learner embedding architecture using only BioBERT, MatBERT, and SciBERT as sublearners. Plots of model loss and F1 scores during training for the multi-learner embedding models are presented in Appendix Figure A3

Architecture	Model variant	Validation F1	Test F1
Voting	Full majority voting	$0.883_{\pm 0.0054}$	0.889
	Full likelihood maximization	$0.884_{\pm 0.0058}$	0.890
	Full boosting	$0.876_{\pm 0.0085}$	0.883
	Ablated majority voting	$\mathbf{0.914}_{\pm 0.0057}$	**0.922**
	Ablated likelihood maximization	$\mathbf{0.914}_{\pm 0.0060}$	**0.922**
	Ablated boosting	$0.911_{\pm 0.0054}$	0.921
Multi-learner embedding	Feed-forward	$0.854_{\pm 0.0059}$	0.904
	BiLSTM	$\mathbf{0.855}_{\pm 0.0110}$	**0.916**
Baseline		$0.848_{\pm 0.0118}$	0.915

Table 5: Performance of ensembled models for NER on Matscholar corpus. Full voting schemes use all five sublearners, while ablated voting schemes use the top-performing subset of three sublearners: MatBERT, BioBERT, SciBERT. Validation results are reported as the mean across 5 trials $\pm 2\sigma$.

We find that all ensemble variants strictly out-perform the baseline model during validation; on the test set, all ablated voting models and the BiLSTM embedding model exceed the baseline. Intriguingly, it is the simplest ensembling method – majority voting over the ablated subset of three learners – that produces the highest validation F1 with least variance across cross-validation trials. This ensemble's 7.8% improvement in validation set F1 score over the baseline far exceeds expectations.

We draw two conclusions from this result. First, transfer learning between transformers with complementary knowledge, even when this knowledge is weakly finetuned to the specific NER task (i.e., constraints on finetuning epochs), can achieve stronger performance than that of a single specialized learner more extensively tuned to the NER corpus. Second, our initial assumption that allowing the ensemble to adaptively reweight the contributions of each sublearner would enhance performance was not validated. The boosted voting and multi-learner embedding architectures – both of which were constructed to allow the ensemble to account for the domain-specific strengths or deficiencies of each sublearner when making predictions – exhibited weaker performance than the simpler majority and likelihood maximization voting schemes. We examine this behavior in the following section.

6 Analysis

We structure our analysis by examining two themes: 1) the reason for the stronger performance of "simple" ensemble schemes (majority voting and likelihood maximization) over other architectures and 2) error analysis for these best-performing models.

At a high level, the boosted voting and multi-learner embedding ensembles operate by learning weights to best aggregate the predictions of each sublearner: this takes the form of the boosting weights α_L^T for the boosted ensemble and the ensemble weights $\mathbf{w}$ for the multi-learner embedding. We visualize the learned weights for three architectural variants in Figure 2

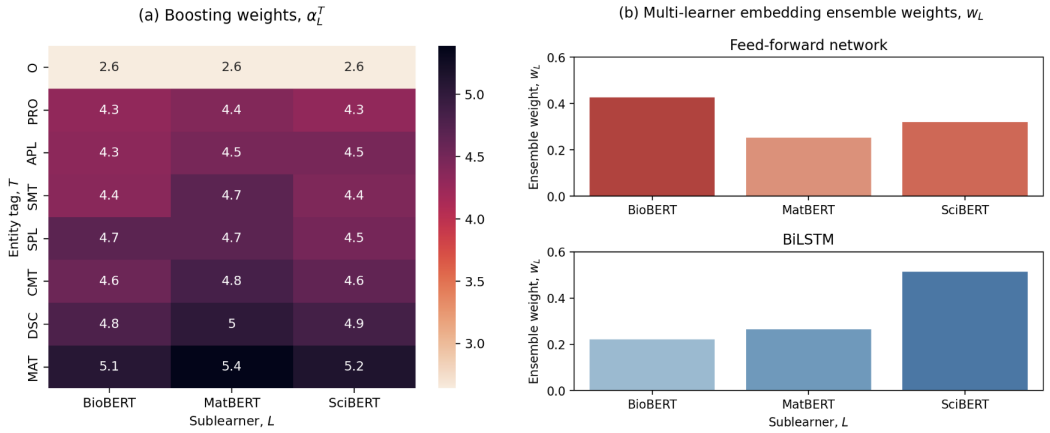

Figure 2: Learned weights for ensemble aggregation in (a) boosted voting architecture and (b) feed-forward and BiLSTM multi-learner embedding architectures

For boosting, we observe a surprising result: the boosting weights α_L^T for any single entity tag T are near-identical across all learners L. This suggests that all three sublearners are similarly "confident" in their predictions for any given tag T; that is, the domain expertises of the sublearners do not significantly impact their per-tag error rates. The unintended outcome of this fact is that the boosted ensemble will disproportionately predict entity tags with high values of α_L^T for all three learners. Inspecting the per-tag predictions of the boosted ensemble supports this interpretation: the ensemble outputs disproportionately many predictions for the "MAT" and "DSC" entity tags, the two tags with highest boosting weights.

For multi-learner embedding ensembles, we find that the three sublearners have unequal contributions to the final token embedding. In both the feed-forward and BiLSTM architecture variations, one of the three sublearners has disproportionately high weight w_L, representing increased contribution

towards the multi-learning embedding relative to the other learners in the ensemble. This suggests that the metalearner network may "over-rely" on this single sublearner BERT when generating token embeddings, losing out on information encoded by the full ensemble.

We also inspect the outputs of the best-performing majority voting and likelihood maximization ensembles to better understand their errors. Figure 3 plots the ensembles' differences in entity tag distribution relative to the true entity labels. Errors predominantly stem from the over-prediction of outside word "O" tags at the cost of under-predicting the material property ("PRO") and characterization method ("CMT") tags. We qualitatively examine several examples where the PRO and CMT tags are misidentified, finding that the ensemble tends to fail in cases where 1) technical scientific notation is interspersed with natural language or 2) the entity of a token depends on a broad surrounding context. Illustrative examples of these cases are given in Table 6.

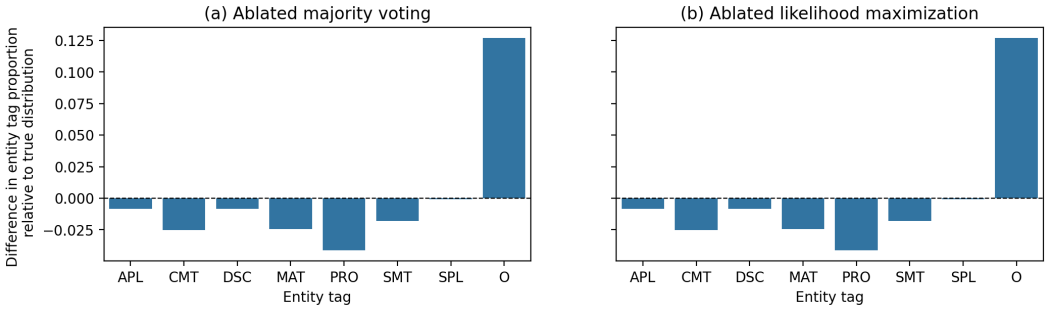

Figure 3: Ensemble models' distribution of entity tag predictions relative to true distribution of entity labels. Both plots share a y-axis representing the difference in predicted to true tag proportion.

Input text	Correct tag	Predicted tag	Commentary
% yield strength	Material property	Outside	The percentage sign is a piece of technical notation with context-dependent meaning: while an isolated % represents a scale of measurement, the phrase "% yield strength" refers to a mechanical property with precise in-domain meaning
***difference** in atomic size*	Material property	Outside	In this setting, the full phrase "difference in atomic size" is treated as a property of a material; the model neglects this broader context and tags "difference" using its single-word definition

Table 6: Qualitative analysis of two common error types for the majority voting and likelihood maximization ensembles. For each example, we consider the entity tag prediction of the bolded word.

The first of the failures may arise from underexposure of the model to technical notation during training. This might be addressed by selectively introducing training examples of such notation during finetuning of sublearners, or by adding a sublearner pre-trained on technical notation standards, such as ChemBERTa (Chithrananda et al., 2020), to the ensemble. The issue of misidentifying context-dependent entities suggests that the sublearner token embeddings do not encode long-range contextual dependencies; this could be addressed by increasing the number of sublearner finetuning epochs while carefully monitoring the variance of the resulting ensemble metalearner.

7 Conclusion

We propose an ensemble-based method inspired by the interdisciplinarity of materials science literature for named entity recognition in the domain. We implement a tokenization alignment algorithm for aligning subword token embeddings across ensemble sublearners with different tokenization schemes. This facilitates the design of two ensemble architectures – a voting-based architecture with three voting scheme variants, and a multi-learner embedding architecture with two network variants.

We find that outputting entity tag predictions as the majority vote across an ablated ensemble of three sublearners achieves a 7.8% improvement in validation multiclass F1 score over the existing state-of-the-art. We note, however, that the model performs poorly in cases where token entities are strongly context-dependent; future studies may investigate how tuning the representation ability of individual sublearners impaxts the overall performance of the ensemble.

References

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text.

Tolgahan Cakaloglu and Xiaowei Xu. 2019. A multi-resolution word embedding for document retrieval from large unstructured knowledge bases.

Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. 2020. Chemberta: Large-scale self-supervised pretraining for molecular property prediction.

Jenny Copara, Nona Naderi, Julien Knafou, Patrick Ruch, and Douglas Teodoro. 2020. Named entity recognition in chemical patents using ensemble of contextual language models.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

_ Explosion Inc. 2023. spacy-alignments: align tokenizations for spacy + transformers. `https://github.com/explosion/spacy-alignments`. Accessed: 2024-2-29.

Trevor J. Hastie, Saharon Rosset, Ji Zhu, and Hui Zou. 2009. Multi-class adaboost. *Statistics and Its Interface*, 2:349–360.

Shu Huang and Jacqueline M. Cole. 2022. Batterybert: A pretrained language model for battery database enhancement.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Sheng-Jie Lin, Wen-Chao Yeh, Yu-Wen Chiu, Yung-Chun Chang, Min-Huei Hsu, Yi-Shin Chen, and Wen-Lian Hsu. 2022. A BERT-based ensemble learning approach for the BioCreative VII challenges: full-text chemical identification and multi-label classification in PubMed articles. *Database*, 2022:baac056.

Eugene Wimberly Myers. 1986. An o(nd) difference algorithm and its variations. *Algorithmica*, 1:251–266.

Elsa A. Olivetti, Jacqueline M. Cole, Edward Kim, Olga Kononova, Gerbrand Ceder, Thomas Yong-Jin Han, and Anna M. Hiszpanski. 2020. Data-driven materials research enabled by natural language processing and information extraction. *Applied Physics Reviews*, 7(4):041317.

Recobo.ai. 2023. Model: chemical-bert-uncased. `https://huggingface.co/recobo/chemical-bert-uncased`. Accessed: 2024-2-29.

Yu Song, Santiago Miret, and Bang Liu. 2023. Matsci-nlp: Evaluating scientific language models on materials science language tasks using text-to-schema modeling.

Amalie Trewartha, Nicholas Walker, Haoyan Huo, Sanghoon Lee, Kevin Cruse, John Dagdelen, Alexander Dunn, Kristin A. Persson, Gerbrand Ceder, and Anubhav Jain. 2022. Quantifying the advantage of domain-specific pre-training on named entity recognition tasks in materials science. *Patterns*, 3(4):100488.

L. Weston, V. Tshitoyan, J. Dagdelen, O. Kononova, A. Trewartha, K. A. Persson, G. Ceder, and A. Jain. 2019. Named entity recognition and normalization applied to large-scale information extraction from the materials science literature. *Journal of Chemical Information and Modeling*, 59(9):3692–3702. PMID: 31361962.

A Appendix

Algorithm A1 AlignEmbeddings

Require: ℓ sublearners to be aligned
nowAligned $\leftarrow$ Stack(L_1)
toAlign $\leftarrow$ Queue($L_2, \ldots, L_\ell$)
completedAlignments $\leftarrow$ 0
while toAlign $\neq \emptyset$ **do**
  prevAlign $\leftarrow$ Peek(nowAligned)
  currAlign $\leftarrow$ Pop(toAlign)
  alignments $\leftarrow$ myerAlign(currAlign, prevAlign)[*]
  for idxToAlign in alignments **do**
    if indices not aligned **then**
      duplicateEmbeddings(currAlign, idxToAlign)
    end if
  end for
  if completedAlignments < $\lfloor \frac{\ell}{2} \rfloor$ **then**
    toAlign.Enqueue(Pop(nowAligned))
  end if
  aligned.Push(currAlign)
  completedAlignments $\leftarrow$ completedAlignments + 1
end while
[*] myerAlign: Returns the indices of tokens in prevAlign that correspond to subword tokenizations of a single token in currAlign

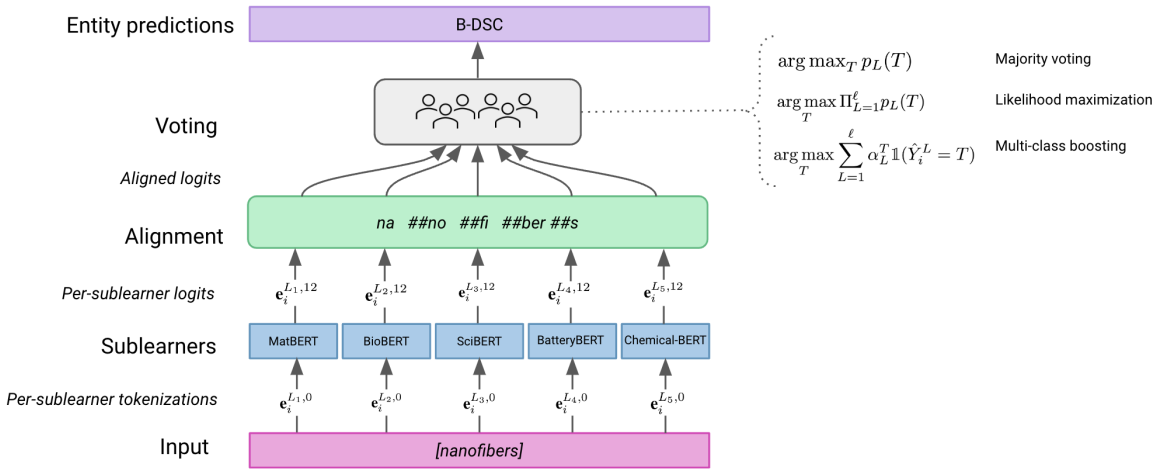

Figure A1: Architecture of the voting ensemble

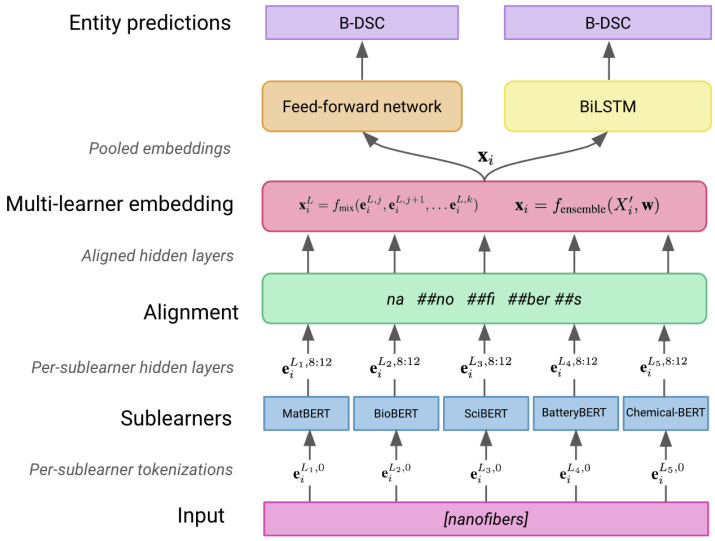

Figure A2: Architecture of the multi-learner embedding ensemble

Sublearners	Majority voting validation F1	Max likelihood validation F1	Boosting validation F1
{MatBERT, BioBERT, SciBERT, Chemical-BERT, BatteryBERT}	0.883	0.884	0.876
{MatBERT, BioBERT, SciBERT, BatteryBERT}	0.908	0.913	0.910
{MatBERT, BioBERT, Chemical-BERT, BatteryBERT}	0.890	0.895	0.893
{Chemical-BERT, BioBERT, SciBERT, BatteryBERT}	0.881	0.884	0.881
{MatBERT, BioBERT, SciBERT, Chemical-BERT}	0.875	0.881	0.879
{MatBERT, Chemical-BERT, SciBERT, BatteryBERT}	0.682	0.697	0.699
{MatBERT, SciBERT, BioBERT}	**0.914**	**0.914**	**0.911**
{MatBERT, BioBERT, BatteryBERT}	0.911	0.912	0.910
{BioBERT, SciBERT, BatteryBERT}	0.908	0.909	0.906
{MatBERT, Chemical-BERT, SciBERT}	0.882	0.885	0.881
{BioBERT, Chemical-BERT, BatteryBERT}	0.879	0.884	0.878
{MatBERT, Chemical-BERT, BioBERT}	0.873	0.879	0.871
{MatBERT, Chemical-BERT, SciBERT}	0.839	0.837	0.835
{Chemical-BERT, SciBERT, BatteryBERT}	0.807	0.815	0.807
{MatBERT, SciBERT, BatteryBERT}	0.751	0.754	0.746
{MatBERT, Chemical-BERT, BatteryBERT}	0.720	0.731	0.717

Table A1: Performance of ablated ensemble models with voting architecture; best performance for each voting scheme is bolded.

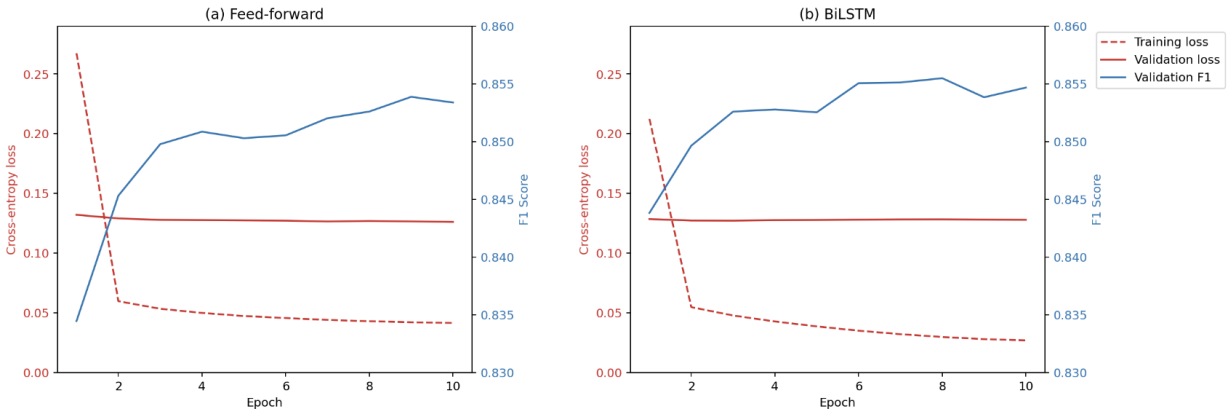

Figure A3: Training curves of feed-forward network and BiLSTM network with ensembled token embeddings as input. The clear plateau in both training validation loss motivated the choice to train for only 10 epochs.