Parameter Tuning for NER Tasks

John Neilson and John Bean

**Abstract**

This work explores hyperparameter tuning for Named Entity Recognition (NER) models using a dataset focused on COVID-19 research papers. We investigate the effectiveness of fine-tuning pre-trained models like spaCy on the CORD-NER dataset, a collection specifically annotated for biomedical entities. The goal is to optimize the NER model's performance in identifying relevant entities within COVID-19 research literature. This paper investigates the effects of changing the learning rate and the batch size of a spaCy model in order to improve performance and decrease training time.

**1 Introduction**

Named entity recognition (NER) models are key components of natural language processing (NLP). They break up raw text by tagging tokens as different entities such as person, place, company, and time. NER models can be especially helpful for filtering through large libraries of information, like during the height of the COVID-19 pandemic when swathes of relevant publications were entering circulation. While many NER models have been fine tuned for general use, their tagging accuracy can be limited when it comes to specialized use cases.

We have studied the impact that different parameters have on a NER model trained on COVID-19 research corpus. We have investigated the impact of learning rate, batch size, and epochs on the accuracy, precision, recall, and F1-Score of our NER model. The purpose of this work is to finetune a model to tag COVID-19 research corpus while determining which parameters result in the best model. We hope that our research into tuning parameters can be applied to other NER tasks within the COVID-19 space and contribute to the performance of NER models.

**2 Related Works**

**2.1 Work 1**The first work is an article titled *NER Hyperparameter Tuning for Finance* which dives into the same idea as this project, but for a different application. They tested changing the learning rate and the dropout of the models in order to try to produce better performance for the models. They used a Roberta-base model for the tasks and then used Bayesian Optimization techniques for the tuning.

**2.2 Work 2**

The second related work is a Github repository discussion post about using a spaCy model and tuning the parameters. They were attempting to modify the same parameters as we did and were running on a google cloud platform. They were also more interested in skipping running the code from the CLI and instead having an executable file to do the hyperparameter tuning.

**3 Methodology**

In our model selection process, we opted for spaCy over the more prevalent BERT model for several compelling reasons. A key factor influencing this decision was the fortunate discovery of a pre-annotated dataset specifically designed for spaCy. Utilizing pre-annotated data streamlines the training process, saving valuable time and resources. Furthermore, spaCy's user-friendly API significantly simplifies model customization and hyperparameter tuning. This ease of use is particularly advantageous when building and iterating on multiple models during the exploration phase.

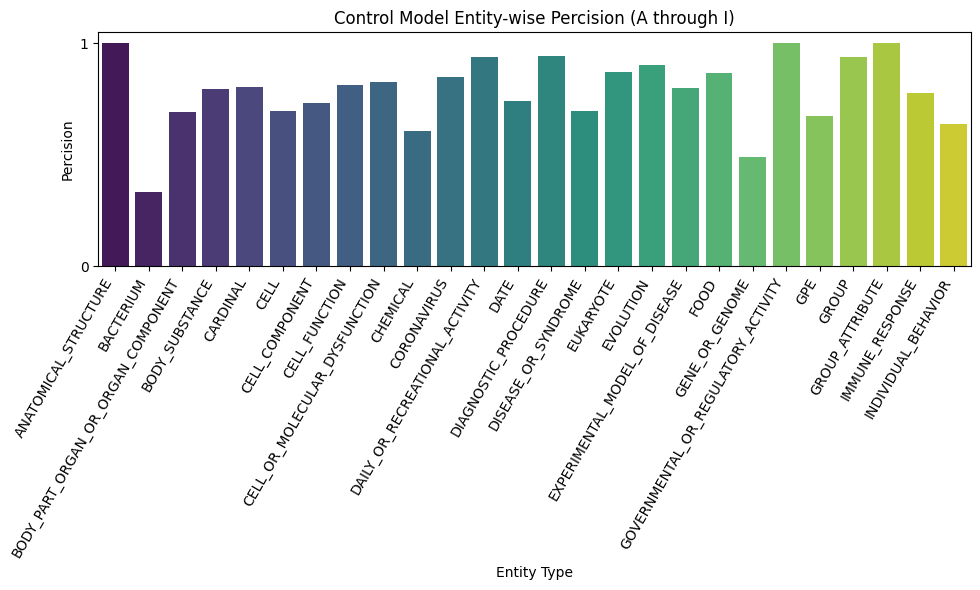
While BERT boasts impressive capabilities, spaCy's streamlined approach, coupled with the availability of pre-annotated training data, proved to be the optimal choice for our specific project. This decision not only expedited the development process but also ensured we could effectively leverage the strengths of a well-established NLP library with a focus on tasks like named entity recognition. In our case we made three different models for testing hyperparameters.

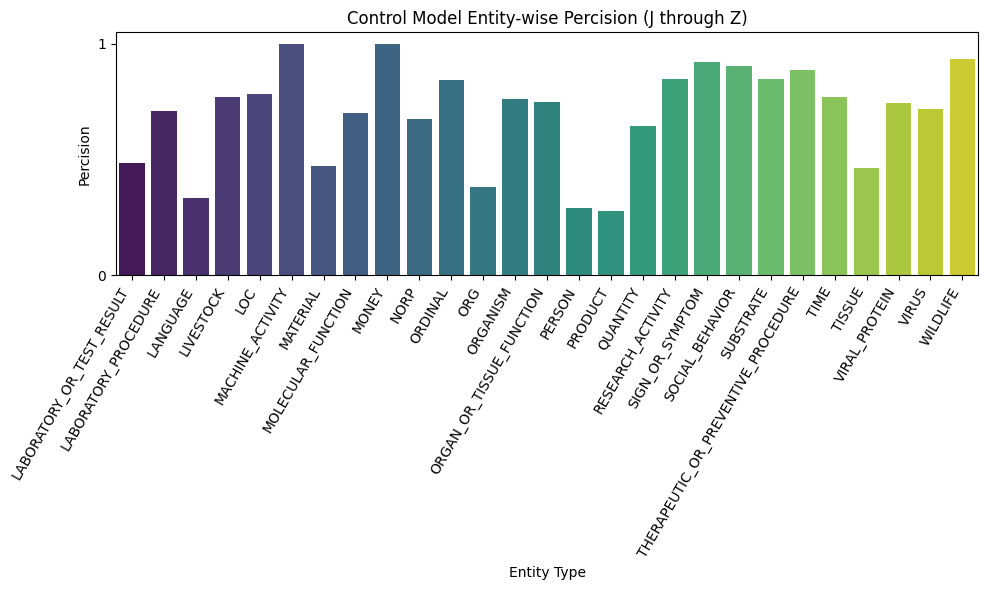
Prior to running the models we had to do a small amount of data reformatting and then due to our own hardware limitations we cut out a large amount of the set in order to be able to train the models. This did affect our overall scores and results but not to the point where the research was useless. The original dataset was roughly 26000 different articles and by the time we had slimmed it down it was roughly 8000.

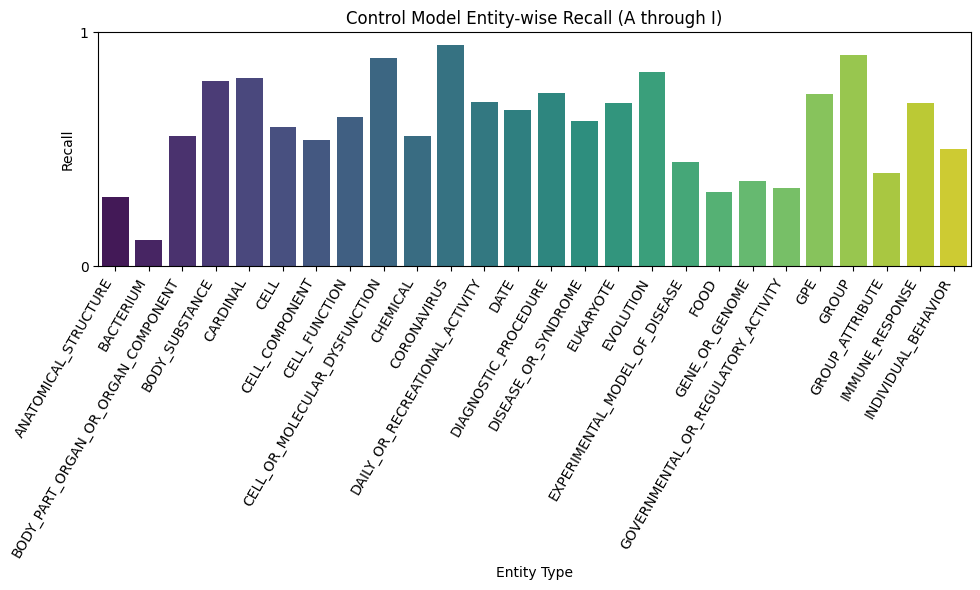
We first ran a base control model using the default parameters assigned by the API that spaCy provides and stores in a configuration file. Next we modified that aforementioned configuration file to increase the learning rate from 0.001 to 0.01 to see if it was able to create a reasonable speedup for training without losing too much accuracy. The last model was where we changed the batch size for the model training from 1000 to 500 to see if it sped up training or improved results.

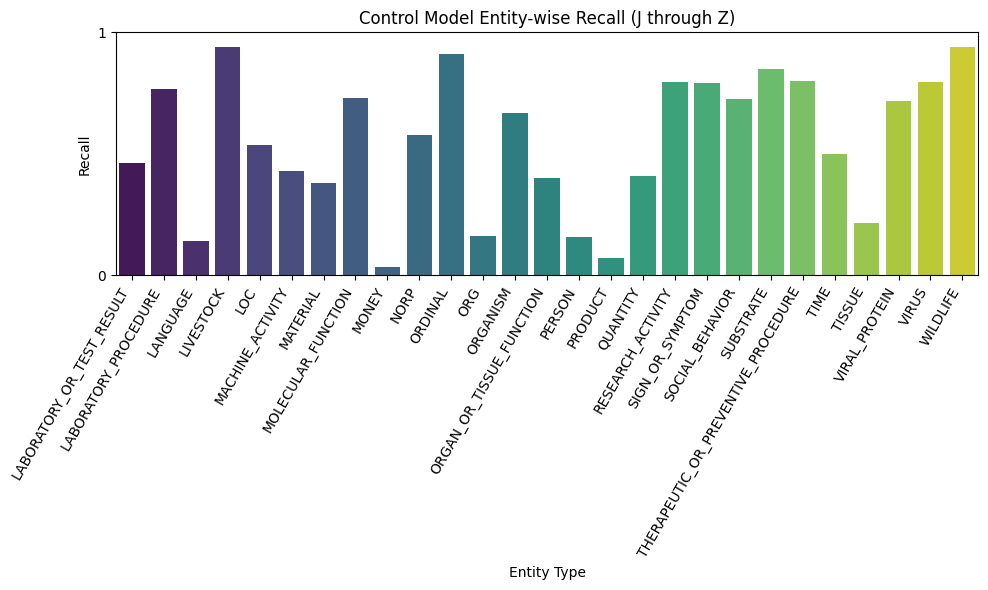
**4 Results & Discussion**

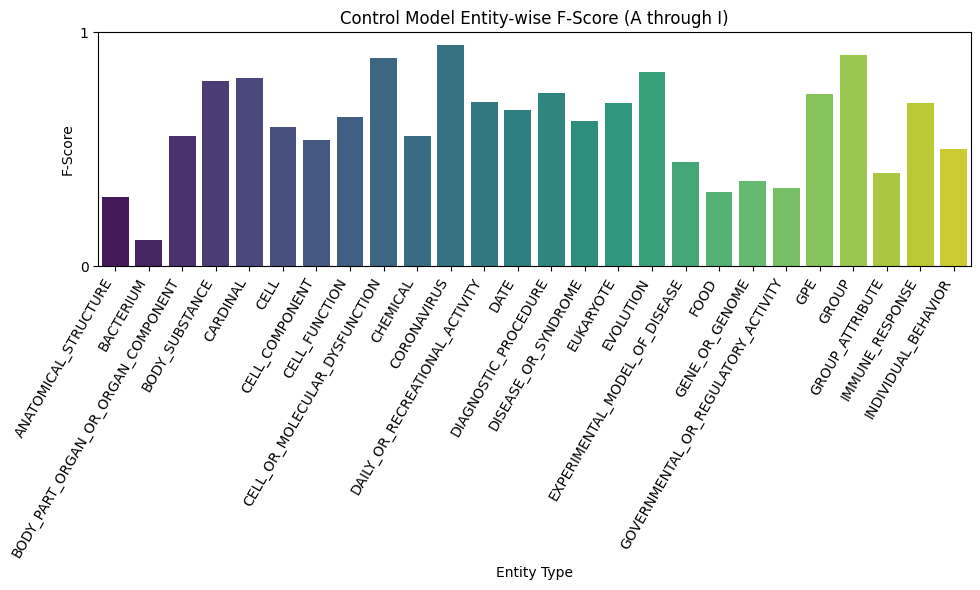
All of the models had similar results when it came to the results of precision tests, which show how much that can just depend on the base model type you work on. For both Recall and F-Score metrics the Control and Decreased Batch performance was identical showing that the decrease in batch size during training did not make much difference. However on both of those metrics the increased learning rate model suffered having worse overall results than the control model, which shows the effects of making a learning rate too large.

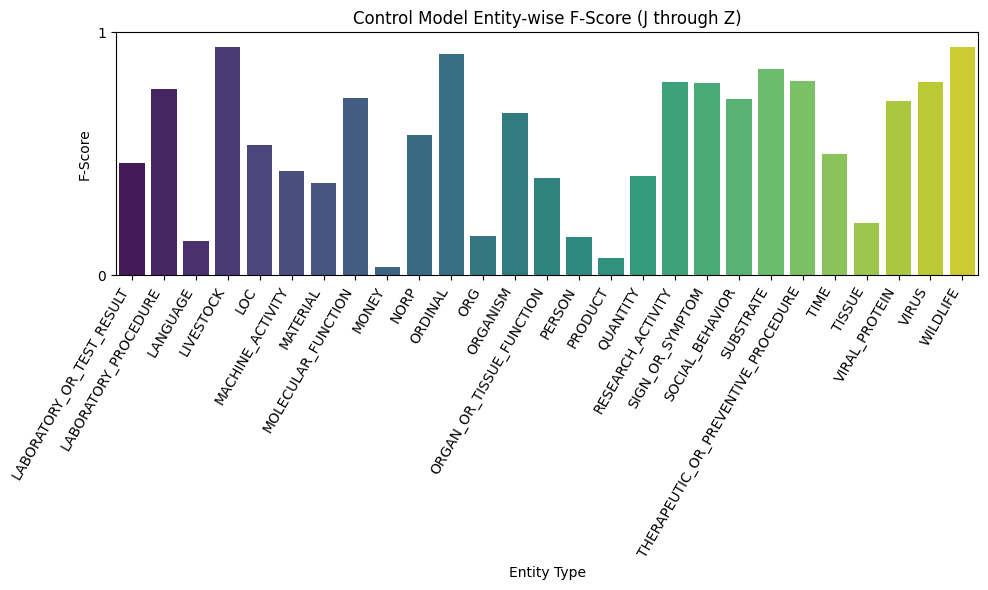




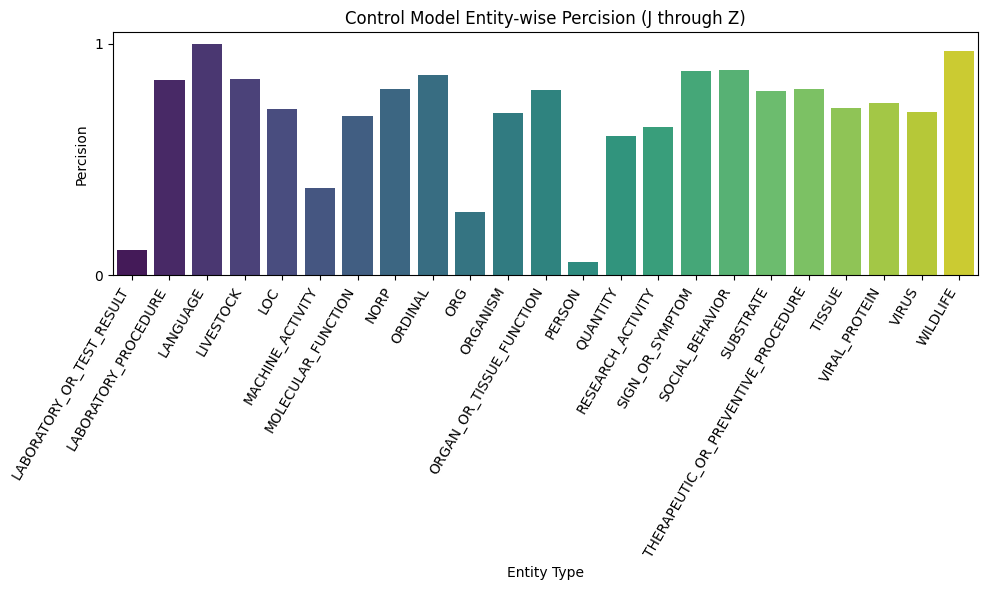


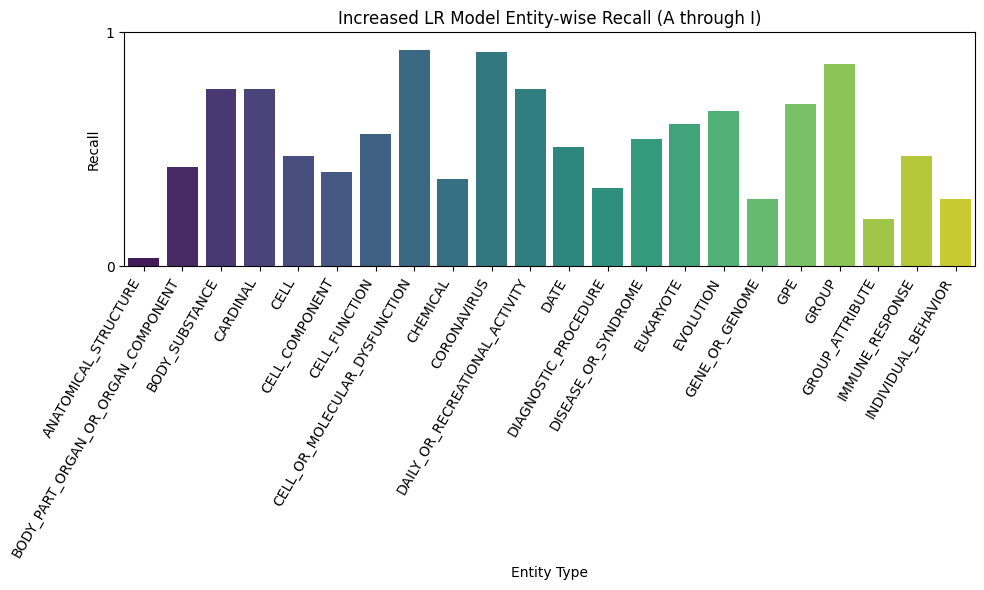


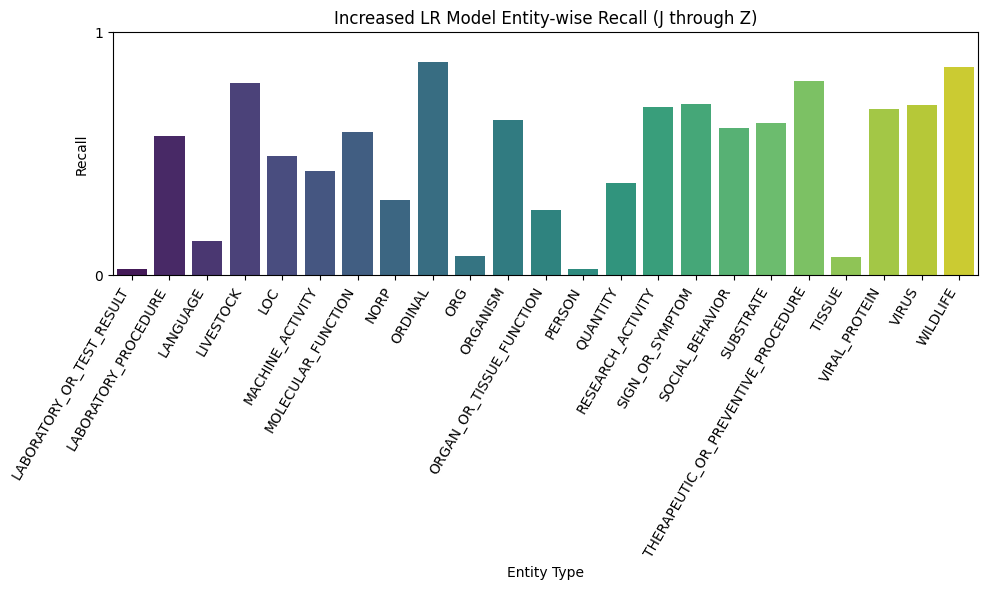


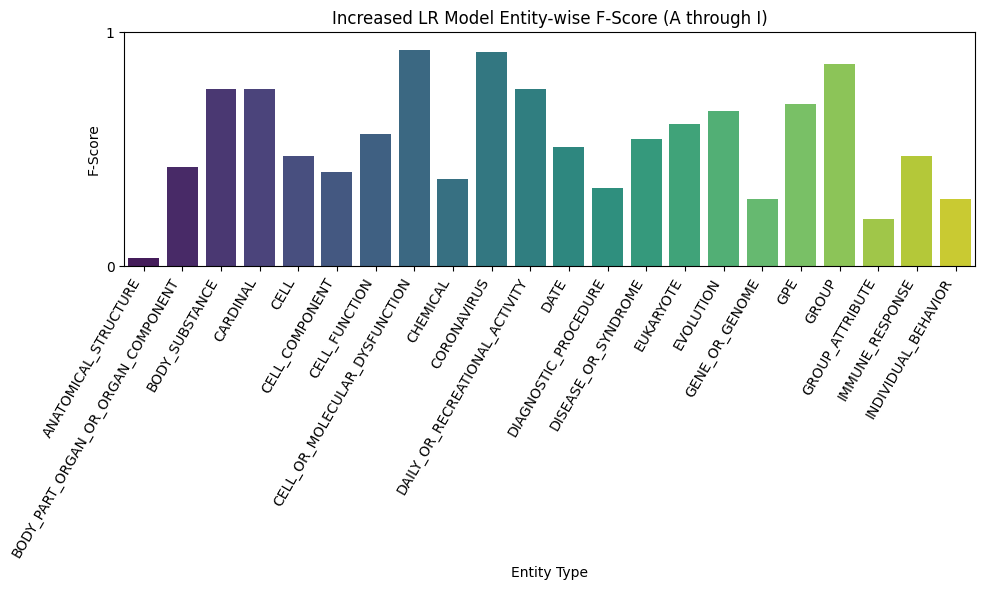
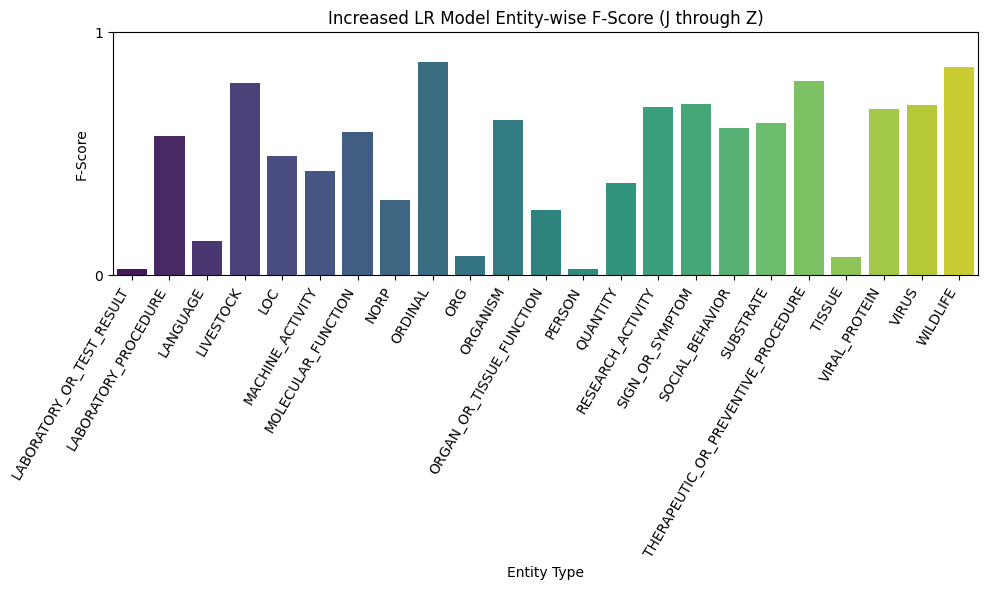


The control model (model 1) yielded mean, median, and mode precision metrics of 0.743, 0.772, and 1.0. The control model (model 1) yielded mean, median, and mode recall metrics of 0.5772, 0.620, and 0.4.The control model (model 1) yielded mean, median, and mode f-score metrics of 0.6266, 0.6569, and 0.5.

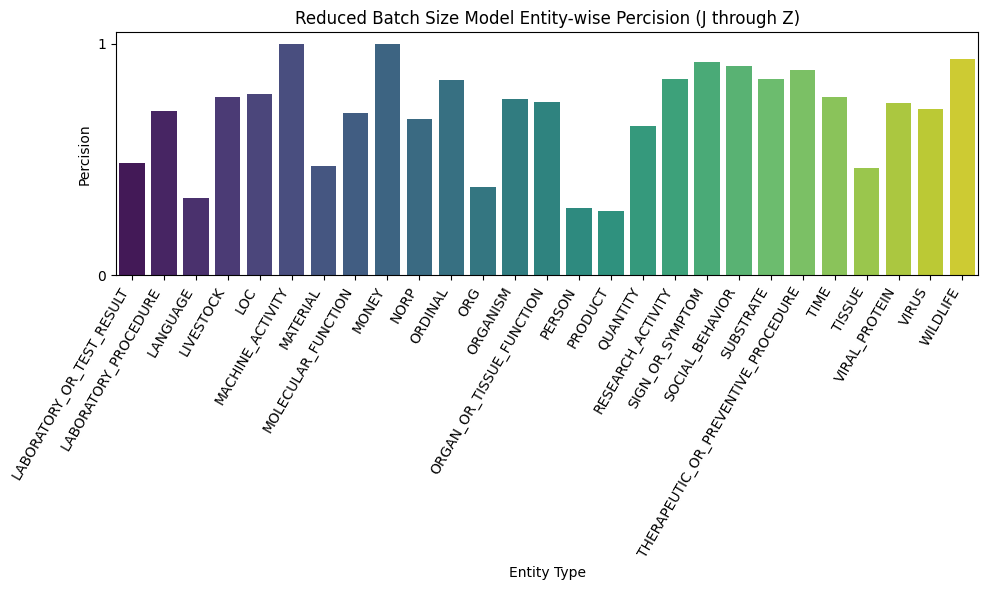


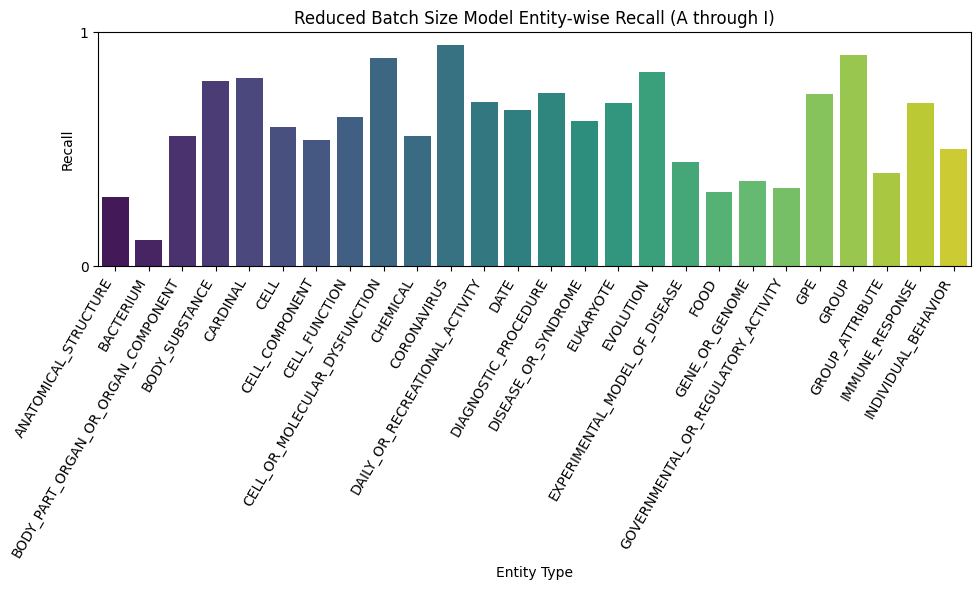


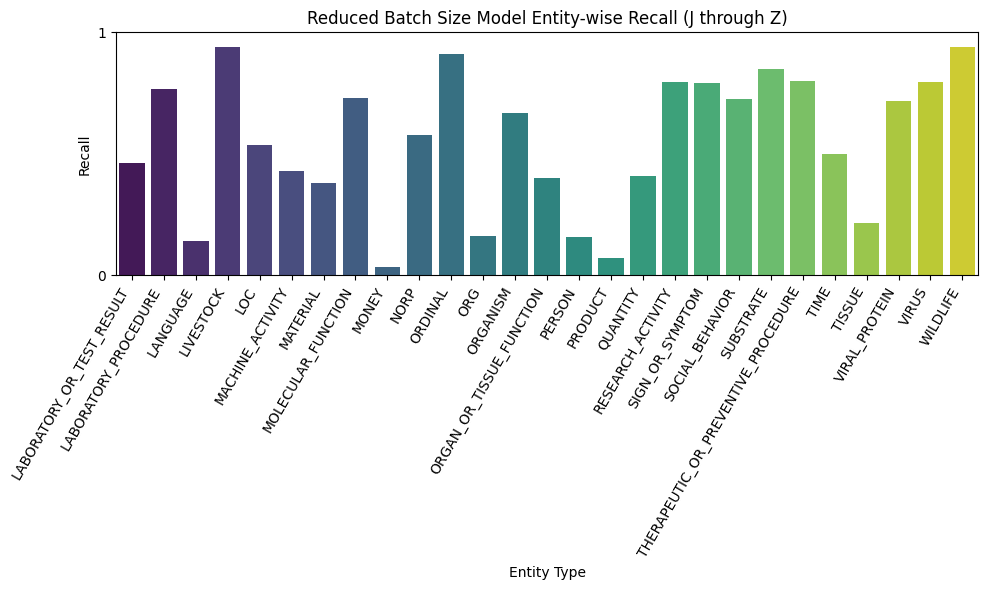


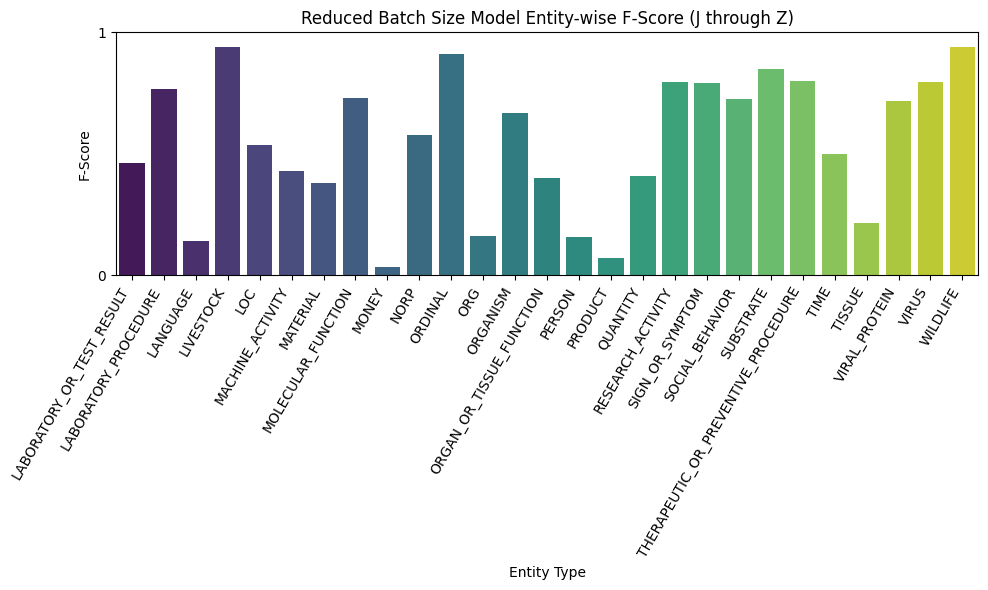
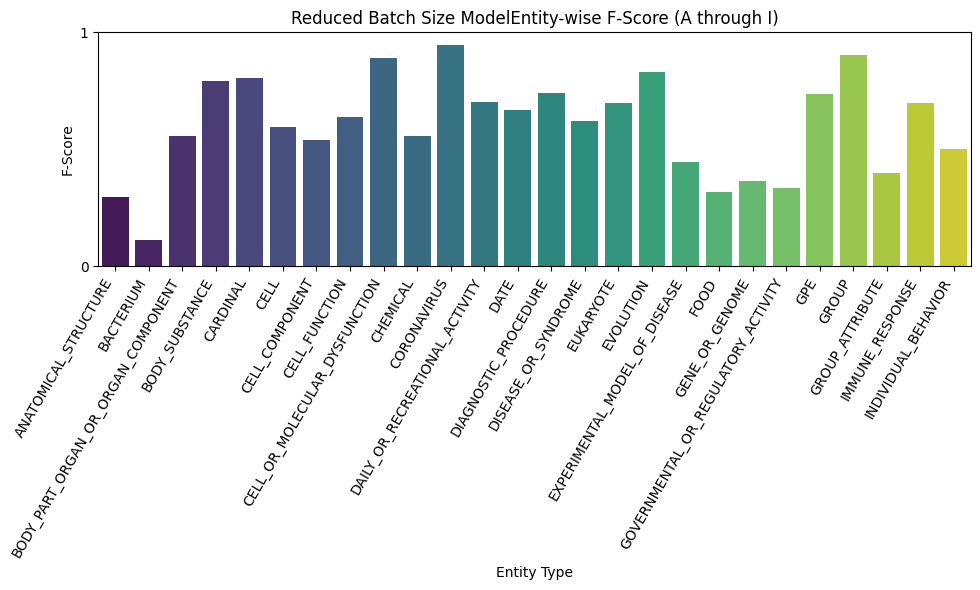
 

The increased learning rate model (model 2) yielded precision metrics of 0.720, 0.762, and 1.0. The increased learning rate model (model 2) yielded recall metrics of 0.516, 0.567, and 0.02439. The increased learning rate model (model 2) yielded f-score metrics of 0.5677, 0.6257, and 0.4.









The decreased batch size model (model 3) yielded precision metrics identical to the control. The decreased batch size model (model 3) yielded recall metrics identical to the control. The decreased batch size model (model 3) yielded f-score metrics identical to the control.

Based on the findings from our research, our next steps are to investigate if these parameters improve the accuracy of models for other NER tasks. We would like to look into NER models for general research corpus, as well as more concentrated–albeit broader than just COVID-19–fields of corpus such as biotechnology or machine learning. We also investigated parameters using spaCy’s NER architecture, so applying our findings to other NER architectures could provide more insight on how architecture impacts the ideal parameters of a model.

We would also like to investigate how changing the batch size impacts the speed of model training, as it does not appear to impact the accuracy of the model in any way. Therefore, if certain batch sizes result in more efficient models, that could be impactful in the NER field.

**5 Conclusions**

Based on our findings we can conclude that changing the learning rate does have an impact on the quality of the model. We found that for NER tasks based on COVID-19 research corpus, lower learning rates yielded higher metrics across the board. We believe that this is because the model has more time with training set and as a result had a better grasp on the research corpus.

We can also conclude that changing the batch size does not have any impact on the quality of the model for NER tasks based on COVID-19 research corpus, as the metrics were identical across the board. This is something that we should look further into, as it could have implications in model efficiency.

**References**

[1] <https://www.kaggle.com/datasets/sushilkumarinfo/covid-ner-data-set/data>

[2] <https://spacy.io/>

[3] <https://wandb.ai/nikhilgavin02/skills_tune_fin_v2/reports/NER-Hyperparameter-Tuning-for-Finance--VmlldzoxOTg2MDE2>

[4] <https://github.com/explosion/spaCy/discussions/11126>