AI/ML for Archive Ingest

At the end of 2023, IPAC hosted groups from JPL and Caltech campus to present on artificial intelligence and machine learning research and development that each has been doing, and explore opportunities for collaboration. The group agreed that there are areas of overlap between usage of AI/ML in projects at JPL and some of the applications at IPAC. IPAC is looking for specific ways to apply these new technologies to improve the efficiency of archival ingestion and perhaps even provide additional analysis capabilities to the science community.

As an initial probe of these technologies for archive ingest, IPAC launched a short-term Natural Language Processing (NLP) task to assess the feasibility of identifying and extracting object names, coordinates, and potentially other data such as redshifts, from the published literature — either from PDF or from HTML available versions of the published articles. Large Language Models (LLMs), which are advanced AI/NLP systems built on neural network architectures like Transformers [1], excel in tasks such as translation, summarization, and content generation, and can be trained for specific applications like Table Question Answering (TQA) or Named Entity Recognition (NER) [2], aligning well with our targeted objective.

For feasibility assessment here, we fine tuned pre-trained LLMs for a NER task (All codes available on our github page [3]). Fine-tuning an existing LLM rather than training one from scratch is advantageous primarily due to resource, time, and data efficiencies [4]. Training LLMs from scratch demands substantial computational resources, often requiring powerful GPUs or TPUs running continuously for weeks or months, and can incur significant costs, sometimes amounting to millions of dollars. In contrast, fine-tuning uses a fraction of these resources and can be completed in days or even hours, depending on the task. The initial training of an LLM also requires enormous datasets, typically involving billions of words or more, while fine-tuning can achieve effective results with much smaller datasets, sometimes only needing thousands or tens of thousands of examples. Pre-trained LLMs come with a broad base of language understanding and general knowledge, which fine-tuning leverages to adapt the model to specific tasks, thus bypassing the need to develop these foundational capabilities from the ground up. This approach not only makes cutting-edge AI technology more accessible to a wider range of users but also reduces the risks associated with model training, such as data quality issues or architectural inefficiencies. Moreover, pre-trained models have often undergone rigorous testing and improvements, ensuring a level of robustness and reliability that freshly trained models might lack.

Existing LLMs like BERT (Bidirectional Encoder Representations from Transformers) [5] and RoBERTa (A Robustly Optimized BERT Pretraining Approach) [6] represent significant advancements in NLP. These LLMs operate on an encoder/decoder architecture, enhanced by Transformers self-attention mechanisms. The self-attention, at its core, allows a model to weigh the importance of different parts of the input data (such as different words in a sentence) when processing each part. In the context of a sentence, for example, self-attention calculates the relevance of all other words to each word in the sentence. It does this by computing a set of attention scores, which determine how much focus to put on other parts of the input when understanding a specific part. The encoder processes the input text using self-attention to understand the context and relationships between words. The decoder then generates output, informed by the encoder's context and its own self-attention, to produce coherent and relevant text. This combination allows LLMs to effectively handle complex language tasks by maintaining a deep contextual understanding throughout the model. BERT, developed by Google, uses a bidirectional approach to understand the context of a word from both left and right sides in a sentence. Its base version has about 110 million parameters, while BERT Large has about 340 million. RoBERTa, an iteration by Facebook AI, modifies key hyperparameters in BERT, removing the next-sentence pretraining objective and training with larger mini-batches and learning rates. These models excel in NER tasks due to their deep understanding of contextual relationships within text, making them adept at identifying and classifying named entities. When choosing an LLM for fine-tuning, it is important to consider factors such as the size of the model (larger models may yield better results but require more computational resources), the nature of the dataset (including domain-specific language), and the specific requirements of the NER task. Additionally, the availability of pre-trained models in the desired language and the adaptability of the model to the specific nuances of the task at hand are crucial considerations. A couple of models fine tuned on astronomical literature already exist (e.g., AstroBert [7], AstroLlama [8]) however they are not suitable for our purpose here as they are not NER models. In this exercise we fine tuned two models, a bert-base-ner [9] and a roberta-large-ner-english [10], both trained on CoNLL2003 dataset [11]. In the future and with more resources, we need to consider and compare other models as well as training/fine tuning a non NER model trained on astronomy literature for a NER task.

Perhaps the most critical step of fine tuning a NER model is preparing its training data. Initially, it is essential to gather a comprehensive dataset relevant to the domain of interest. Here, we gathered the Astrophysical Journal extragalactic publications of the 2020s which include our expected range of linguistic contexts and styles. Text from these publications is extracted from their HTML files with BeautifulSoup [12], a python library for parsing structured data. The next step is annotation, where entities in the text are labeled accurately for a NER task. This process requires meticulous attention to detail, as the quality of the annotations directly impacts the model's performance. The entities to be labeled can vary depending on the domain, for our task, we included object_names, coordinates, and redshifts. It's crucial to define a consistent labeling scheme. The BIO (Beginning, Inside, Outside) format is commonly used in LLMs [13]. In this format, "B" labels the beginning of an entity, "I" labels the inside of an entity, and "O" indicates tokens outside any entity. For example, in "San Francisco is a city", "San" would be tagged as B-LOC and "Francisco" as I-LOC (where LOC stands for location as an entity in NER), and every other word in the sentence as O. Instead of manual annotation for this task which can be time-consuming, we used the tables from the NASA Extragalactic Database (NED) archives which were human extracted over the years and only converted their format to be consistent with our models BIO labeling scheme. Significant improvements can be made in this area, such as labeling variants of the same entity, for example, “NGC-6946”, “NGC 6946”, and “NGC 6946”. After annotation, the data must be preprocessed for compatibility with the LLM. This involves tokenizing the text, converting it into a format understandable by the model, and possibly balancing the dataset to avoid biases towards certain entity types. Finally, we reserved 20% of the data for validation and testing, allowing for the evaluation of the model's performance during and after the fine-tuning process. This careful preparation of training data is vital for effectively fine-tuning an LLM for NER tasks, as it ensures the model learns to identify and categorize entities accurately in the context of the specific application.

In this proof of concept, we leveraged the Hugging Face Transformers library [14] to access the pre-trained models mentioned above and for the fine-tuning of our LLM, we adopted Parameter-Efficient Fine-Tuning (PEFT) [15], an approach that optimizes the tuning process by adjusting a minimal subset of model parameters, making it ideal for large-scale models. Additionally, to monitor and manage the fine-tuning process, we integrated Weights & Biases (wandb) [16], a powerful experiment tracking tool. Wandb proved crucial for logging, visualizing, and comparing various model runs, allowing us to track critical metrics such as loss and accuracy in real-time (see Figures). The screenshot below illustrates applications of our trained NER model for inference purposes. The first example demonstrates the identification of entities in a text array, presented in the BIO format. The second example showcases the model's capability to process an HTML document from a paper, specifically to query object names.

```
text = "Groundbased data for the other two low-metallicity clusters in our sample, NGC 6809 ([Fe/H] = -1.80)\
and NGC 5139 ([Fe/H] = -1.62), are also reproduced reasonably well by the [Fe/H]mod = -1.65 isochrones at an\
age of 13 Gyr (Figs 7b and 7c). As in the case of NGC 6397, the models do not reproduce the full extension of\
the blue horizontal branch. These objects are at z=0.012 and not very star forming."

pred = nlpeft(text)
print(format_pred_for_print(pred,text,conf=0.9))
```

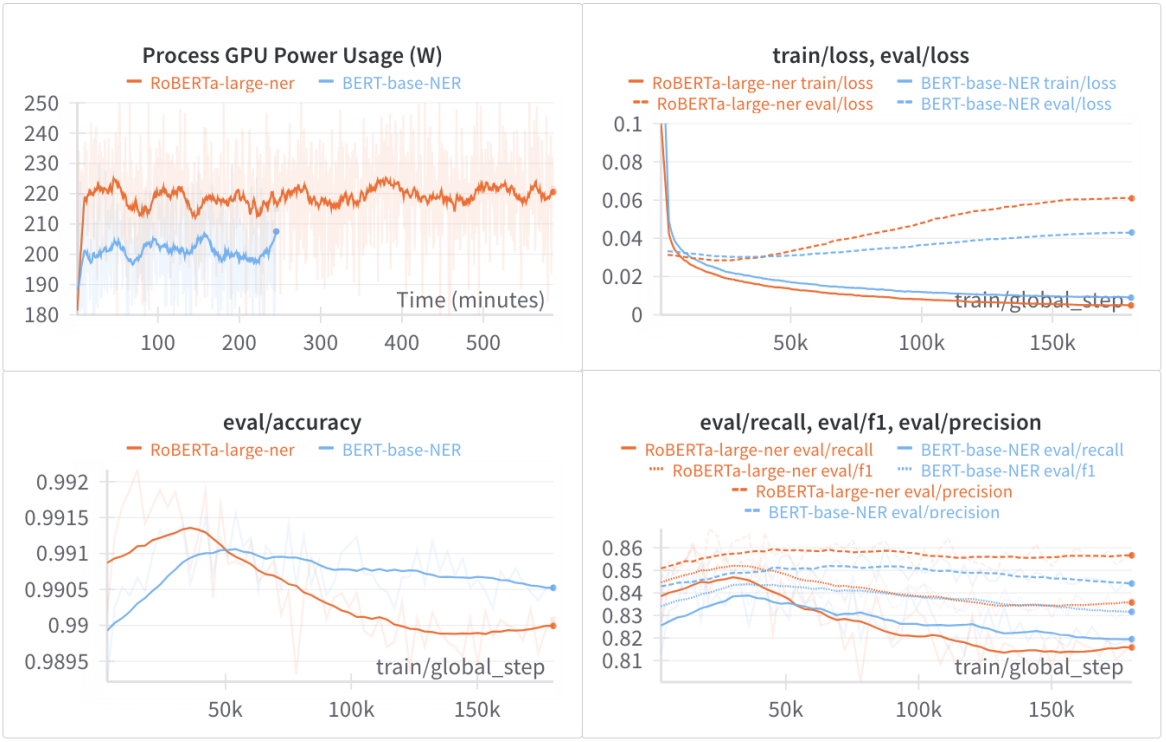
```
Groundbased data for the other two low-metallicity clusters in our sample, [N (B-name 1.00)][GC (I-name 0.99)]
[680 (I-name 1.00)][9 (I-name 1.00)] ([Fe/H] = -1.80)and [N (B-name 1.00)][GC (I-name 0.97)] [51 (I-name 1.00)]
[39 (I-name 1.00)] ([Fe/H] = -1.62), are also reproduced reasonably well by the [Fe/H]mod = -1.65 isochrones at
anage of 13 Gyr (Figs 7b and 7c). As in the case of [N (B-name 1.00)][GC (I-name 1.00)] [63 (I-name 1.00)][9 (I-
name 1.00)][7 (I-name 1.00)], the models do not reproduce the full extension ofthe blue horizontal branch. These
objects are at z=[0 (B-redshift 0.97)].012 and not very star forming.
```

```
names_in_paper('data/2022-ApJ-Vol925/HTML/2022ApJ...925..182H.html')
```

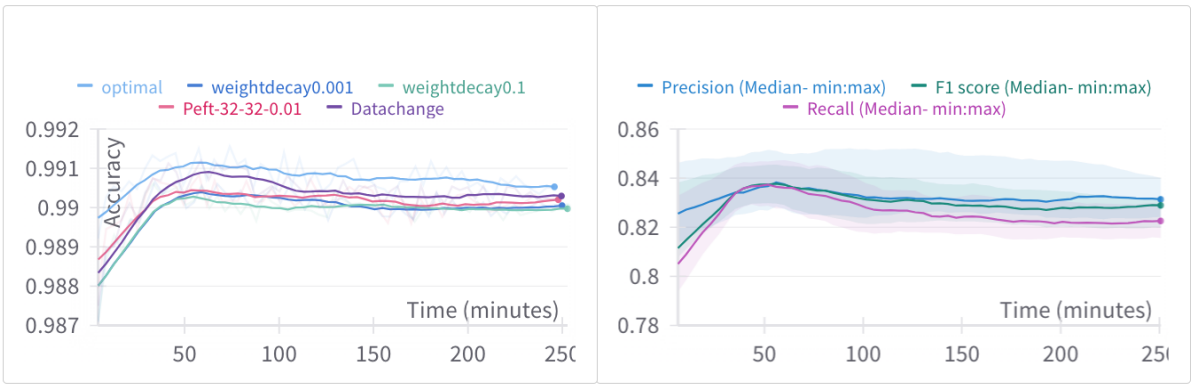
```
['GRB190829', 'GRB1908929']
```

We evaluated the performance of our models using standard ML metrics (i.e., accuracy, precision, recall, and F1 score), each offering a unique insight into different aspects of model effectiveness. Accuracy measures the overall proportion of correct predictions, both positive and negative, made by the model. This is expectedly very high (~99%) for our model since it is easy to identify negatives or non-entities. Precision assesses the accuracy of the model’s positive predictions, indicating the proportion of true positives among all predicted positives. Recall, or sensitivity, evaluates the model's ability to correctly identify actual positives, showing the ratio of true positives to all actual positive instances. The F1 score, a harmonic mean of precision and recall, provides a balanced measure of a model’s precision and recall, particularly valuable in scenarios of imbalanced classes. Our models achieve approximately 80-85% in recall, precision, and F1 score, which suggests that they miss about 20% of the entities. A closer examination of these missed entities reveals that the primary challenge lies in LLMs handling of numerical data. The observed issue primarily stems from the non-continuous tokenization of numbers (i.e., treating them as words) in the models. Recent advancements have proposed alternative approaches to

address this, such as treating numbers differently during the tokenization process, as outlined in recent literature ([17-18]).



In Figure 1, we benchmarked the models fine-tuned from BERT and RoBERTa architectures, reaching 85% precision. We selected the model checkpoint at approximately 50,000 steps for our use, as this represents the point of peak performance. Beyond this stage, the model begins to overfit the data, a trend indicated by the increasing evaluation loss despite a continuing decrease in the training loss. Despite similar performance metrics between the two, we chose the BERT-base model for its operational efficiency, specifically its faster processing time. Figure 2, shows additional hyper parameter tuning on the BERT model, including adjustments of number of layers in PEFT parameters to be trained, adjustment of the model weight decay, as well as some variation in annotation of the training data. These modifications, however, showed minimal impact on the metric as can be seen from the shaded regions of the right panel, indicating the robustness of our model's performance against these particular parameter changes.



In conclusion, our exercise of employing NLP techniques for the identification and extraction of information from astronomical literature serves as a significant proof of concept. This endeavor highlights the potential for evolving these methods into robust production tools within the field. The utilization of LLMs is becoming increasingly accessible and efficient, offering substantial savings in terms of manpower and resources. Notably, the ability of these models to understand context and perform Question Answering (QA) in scientific literature presents immense opportunities. These capabilities are not just facilitative tools but are poised to become building blocks in the development of scientific foundation models.

References

1) Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems* 30 (2017)
2) Nadeau, David, and Satoshi Sekine. "A survey of named entity recognition and classification." *Lingvisticae Investigationes* 30.1 (2007): 3-26
3) Hemmati, Shoubaneh, “NED AI initiative”, 2023, https://github.com/xoubish/NEDAI
4) Ziegler, Daniel M., et al. "Fine-tuning language models from human preferences." *arXiv preprint arXiv:1909.08593* (2019)
5) Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805* (2018)
6) Liu, Yinhan, et al. "Roberta: A robustly optimized bert pretraining approach." *arXiv preprint arXiv:1907.11692* (2019)
7) Nguyen, Tuan Dung, et al. "AstroLLaMA: Towards Specialized Foundation Models in Astronomy." *arXiv preprint arXiv:2309.06126* (2023)
8) Grezes, Felix, et al. "Building astroBERT, a language model for astronomy & astrophysics." *arXiv preprint arXiv:2112.00590* (2021)
9) George, Victor Vadakechirayath. "Neural architecture impact on identifying temporally extended Reinforcement Learning tasks." *arXiv preprint arXiv:2310.03161* (2023)
10) Baptiste, Jean. "Roberta-large-ner-english." Hugging Face, 2023, https://huggingface.co/Jean-Baptiste/roberta-large-ner-english
11) Sang, Erik F., and Fien De Meulder. "Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition." *arXiv preprint cs/0306050* (2003)
12) Richardson, Leonard. "Beautiful soup documentation." (2007)
13) Ramshaw, Lance A., and Mitchell P. Marcus. "Text chunking using transformation-based learning." *Natural language processing using very large corpora*. Dordrecht: Springer Netherlands, 1999. 157-176
14) Wolf, Thomas, et al. "Huggingface's transformers: State-of-the-art natural language processing." *arXiv preprint arXiv:1910.03771* (2019)
15) Ding, Ning, et al. "Parameter-efficient fine-tuning of large-scale pre-trained language models." *Nature Machine Intelligence* 5.3 (2023): 220-235
16) Biewald, Lukas. "Experiment Tracking with Weights and Biases." 2020. Weights & Biases, https://www.wandb.com/
17) Loukas, Lefteris, et al. "FiNER: Financial numeric entity recognition for XBRL tagging." *arXiv preprint arXiv:2203.06482* (2022)
18) Thawani, Avijit, Jay Pujara, and Filip Ilievski. "Numeracy enhances the literacy of language models." *Proceedings of the 2021 conference on empirical methods in natural language processing*. 2021