AI for Data Ingestion into IPAC Archives

Nicholas Susemiehl and Joseph Mazzarella
IPAC, California Institute of Technology, 1200 E. California Blvd., Pasadena, CA 91125, USA

Abstract

The data archives at IPAC, including the NASA Extragalactic Database (NED) and NASA Exoplanet Archive (NEA), have served as repositories for data published in the astronomical literature for decades. Throughout this time, extracting data from journal articles has remained a challenging task and future large data releases will exasperate this problem. We seek to accelerate the rate at which data can be extracted from journal articles and reformatted into database load files by leveraging recent advances in natural language processing enabled by AI. We are developing a new suite of tools to semi-automate information retrieval from scientific journal articles. Manual methods to extract and prepare data, which can take hours for some articles, are being replaced with AI-powered tools that can compress the task to minutes. A combination of AI and non-AI methods, along with human supervision, can substantially accelerate archive data ingestion. Challenges remain for improving accuracy, capturing data in external files, and flagging issues such as mislabeled object names and missing metadata.

1 Introduction

The NASA Extragalactic Database (NED)[1] and NASA Exoplanet Archive (NEA)[2] are two astronomical data repositories operated by IPAC at the California Institute of Technology which have served the scientific community since 1990 and 2011 respectively. NED has collected over 1.1 million distinct objects, including galaxies, quasars, and gamma ray bursts. NEA seeks to provide a complete list of confirmed exoplanets, which now number over 6,000, and their stellar hosts. New data flow through similar pipelines for both NED and NEA as they are prepared for ingestion into the archives' internal databases. Newly-published articles are found via queries to the listing services of the Astrophyiscs Data System (ADS)[3]. These articles are then fed through a relevance classification model, which seeks to predict whether or not the data from an article should be ingested into the archive. A scientist then selects the relevant papers from a user interface displaying the relevance classifier results. Next, the appropriate data is extracted from the article and transformed into the particular load file formats for NED and NEA before being ingested into the databases. Throughout most of the history of these archives, the data extraction and load file creation process has been done manually, largely because astronomical journal articles vary widely in structure and semantics. While this manual process has been functional, both archives currently have backlogs of unprocessed published journal articles and keeping up with newly published literature can be difficult. To add to this, anticipated exoplanet candidate detection yields from future missions have the potential to substantially increase NEA's holdings. Data releases from missions such as Gaia (Perryman et al., 2014), Roman (Penny et al., 2019; Wilson et al., 2023), PLATO (Matuszewski et al., 2023), and Earth 2.0 (Ge et al., 2024) include estimated yields of thousands to hundreds of thousands of candidate exoplanets. Given these realities, it has become important for the data archives at IPAC to enhance the throughput of their data ingestion pipelines.

The field of natural language processing has yielded tools that are increasingly-capable of mining data from the text of scientific journal articles. This work initially investigated Word2Vec (Mikolov et al., 2013) and its extension Doc2Vec (Le and Mikolov, 2014). Word2Vec/Doc2Vec have largely been improved upon by transformer-architecture large language models (LLMs), which use attention mechanisms to create more dynamic

[1] https://ned.ipac.caltech.edu/
[2] https://exoplanetarchive.ipac.caltech.edu/
[3] https://ui.adsabs.harvard.edu/

Proceedings of the Third Workshop for Artificial Intelligence for Scientific Publications, pages 87–93

contextual understanding of text (Devlin et al., 2018; Liu et al., 2019; Brown et al., 2020; Touvron et al., 2023a). LLMs can function as foundation models and be finetuned or directed via prompt engineering to complete downstream tasks.

Other works have shown that foundational LLMs pretrained on domain-specific data (in particular, astronomy - c.f. Grezes et al., 2022, 2024; Bhattacharjee et al., 2024; de Haan et al., 2025) outperform other LLMs not attuned to this domain on downstream tasks related to it. This project therefore uses models pretrained on the astronomical literature. We initially considered the encoder AstroBERT (Grezes et al., 2024, 2022) and decoder AstroLlama (Dung Nguyen et al., 2023), but decided to use the encoder INDUS (Bhattacharjee et al., 2024) and decoder AstroSage-Llama-3.1-8B (AstroSage; de Haan et al., 2025) instead. INDUS and AstroSage are based on more advanced models than the preceding AstroBERT and AstroLlama: astroBERT uses the architecture of BERT (Devlin et al., 2018) and INDUS is based on RoBERTa (Liu et al., 2019), while AstroLlama uses the Llama-2 (Touvron et al., 2023b) architecture and AstroSage is based on that of Llama-3.1 (Grattafiori et al., 2024).

The goal of this work is to produce a tool which accelerates the processes of data extraction and load file creation for NED and NEA. There is no expectation that the load files created using these tools will be perfect, so automated issue flagging, human supervision, and periodic re-training of models will be integral to this process.

2 Methods

A variety of methods, both AI-based and not, are being deployed at the different stages of the archive data ingestion pipeline.

2.1 Data Retrieval

Each module of this work uses the full text of a journal article. These are downloaded from ADS using their API service and converted from the PDF format to plain text using the PyMuPDF loader provided by LangChain[4]. We used the INDUS tokenizer to convert this text into the appropriate format when using INDUS.

[4]https://docs.langchain.com/oss/python/langchain/overview

2.2 Relevance Classification

Both NED and NEA already use machine learning classifiers to predict the probability that an article is relevant to their holdings. The relevance classifier used by NED (Chen et al., 2022a) is based on the Stanford Classifier (Finkel et al., 2005), while the NEA tool (Susemiehl & Christiansen, in prep.) inputs Doc2Vec embeddings to a logistic regression model. Both of these tools have successfully automated this task. However, their accuracy has been declining due to changes in content and structure of newer journal articles, and transformer-based LLMs finetuned to this task have the potential to more accurately predict paper relevance. Due to the active development cycle of this project, this tasks is being reserved for after after the completion of other modules (see Future Work S4.1).

2.3 Data Extraction

Once a relevant article is identified, the data it presents must be extracted into load files that can be ingested into the NED and NEA databases. The data in these articles, such as a planet's mass or a galaxy's redshift, can be contained in the main text or tables within an article, and also in external files linked to some articles. Transformer-based LLMs have as input one-dimensional strings of text, so the two-dimensional structure of tables is lost during training/inference. We therefore employ different methods for text and tabular data extraction.

2.3.1 Object Name Detection

The detection of the names of astronomical objects which are presented in a given article is a fundamental task in this work. To this end, we finetuned INDUS (Bhattacharjee et al., 2024) instances using the HuggingFace Python library (Wolf et al., 2019) on token classification tasks for both archives.

While both NED and NEA hold large databases of object names and their corresponding article identifiers, the locations of the names within these articles is not recorded. In order to frame this task as a supervised learning problem, it is necessary to label each token in an article as either an object or not an object. While human annotation is commonly employed to label training data in similar cases, this is an expensive endeavor. We sought to leverage the large set of object names and articles held by NED and NEA by automatically labeling the tokens within each article. We converted each object name in the NED and NEA lists to generic forms using regular expressions. The formulated

regular expressions allow for variations in separators, abbreviations, numerical digits, and planet letter suffixes from the published object names to the canonical forms in the archives. A challenge of this technique is in eliminating both false positive and false negative labeling, as mislabeled tokens pollute the training data set and limit model performance during inference.

Following the regular expression-based token labeling, the training data sets were composed of 8268 articles (300.4 million tokens) for the NED model and 2230 articles (89.6 million tokens) for the NEA model. A hyperparameter search of 10 trials was performed over the learning rate, dropout, weight decay, and random seed. The INDUS models were then finetuned for 10 epochs using the HuggingFace framework. The NED model achieved an F1 score on an unseen test set of 0.95 while the NEA model scored an F1 of 0.94 on its test set. However, an investigation of the learning curve (Figure 1) reveals that both models failed to learn the validation data. This is corroborated by a high occurrence of incorrect labels while qualitatively assessing the models' performances. We found that this finetuned version of INDUS outperforms the base model on the name identification task, suggesting the finetuning process was still useful. External validation tools, which compare potential names to expected name formats, are used to reduce false positive predictions during usage of the tool. It seems likely that this poor model performance is caused by pollution of the training data set during the labeling phase, so future work will investigate means to improve this process.

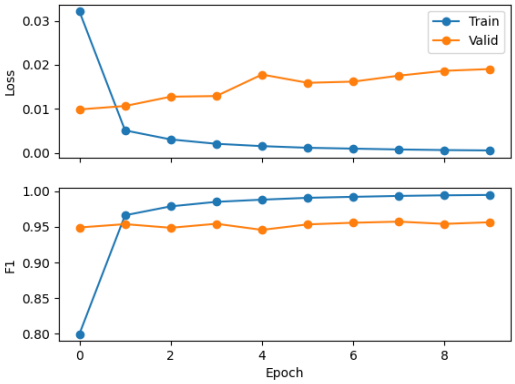

Figure 1: A typical example of a learning curve from finetuning the object name detection model on NEA data (NEA and NED learning curves are qualitatively similar. The flat validation loss curve indicates that the model failed to learn during finetuning.

2.3.2 Text Extraction

Necessary data are often included within the body text of an article. These can be numerical values (e.g. coordinates) or words/phrases (e.g. telescope). Examples of data types regularly found in the body of an article include type of an extragalactic object and the method used to detect an exoplanet. The usage of synonyms, abbreviations, and acronyms for these values is common in the literature. Given the unstructured nature of the body text and the difficulty in composing a token-level training dataset for heterogeneous labels, supervised finetuning approaches may be less applicable. Instead, generative AI is useful because of its ability to read large contexts and answer questions pertaining to data extraction from prompts. We prompt the decoder LLM AstroSage (de Haan et al., 2025) to return the various data types needed to populate a database load file. This is done in zero-shot context without finetuning. Any article longer than the effective context length of AstroSage (Llama-3.1 8B), 32,000 tokens (Hsieh et al., 2024), is broken into sections of 512 tokens each. A retrieval augmented generation system then selects the most relevant 512 token-sized chunks of text to fill the 32,000 token context (i.e. the top ∼62 chunks are combined to serve as the prompt's context). Otherwise, sufficiently short articles are included entirely within the prompt's context. Grammar-constrained decoding is leveraged to force the output of the LLM into a JSON format with keys corresponding to the needed data types. The LLM-outputted values are further required to be chosen from a particular set of possible options for some categorical data types (e.g. object class). These methods control the output to align with the data types expected to be included in the database load files.

2.3.3 Table Extraction

The structured nature of tables is advantageous for extracting data from them. A variety of detected and derived object parameters, such as redshift and mass, are commonly found within the tables of NED and NEA articles. LLMs, while useful for unstructured data (text), reduce the 2-D grid of a table down to a 1-D string which makes the correct alignment between labels and their respective values difficult. Including positional and descriptive tags within the table text included in prompts to the AstroSage was not found to improve the correct

association between extracted values and their respective objects in this work, so methods not solely reliant on an LLM were investigated for this task.

Tables are identified within articles and extracted using the GMFT package[5], which converts the PDF tables to Pandas dataframes (Wes McKinney, 2010). Next, every cell in the dataframe is labeled as either an astronomical object using the object name detection model or a parameter label. We achieve parameter label assignments by matching each as-published potential label to a dictionary of previously-seen labels, converting both to embeddings, and calculating a cosine similarity score. The label type corresponding to the highest matching previously-seen label is then assigned to the as-published label in the table's dataframe (if there is a match score greater than 0.9). With both the objects and parameter labels of a table identified, the dataframe cell containing the respective value is assumed to be the lower-right intersection of the object and parameter label positions in the 2-D grid of the dataframe. This enables direct, automated extraction of data values while maintaining alignment between the object and parameter labels. However, the dictionary of previously-seen parameter labels needs to be expanded whenever substantially different label presentations are encountered.

2.4 Load File Creation

Once data are extracted from the text and tables of an article, they are cleaned and reformatted into database load files using programmatic methods in Python. There are additional components to these files which can be inferred without the use of the above methods.

2.4.1 Other data

NED and NEA load files contain sections of "metadata" regarding the objects to be ingested. This includes, for example, the addition of aliases for a given object. The aliases which need to be added for an object can be inferred by comparing existing entries in the NEA database to those in external databases (e.g. Simbad). Other metadata, like the internal updates to a system's orbital configuration, can be inferred by querying of the NEA databases. An example of inferrable data for NED is the coordinate system (sexagesimal or decimal degree), which can be ascertained via regular expression matching of the retrieved coordinate value.

[5]https://github.com/conjuncts/gmft

3 Results

Prototype versions of these tools have been developed to enable the creation of a database load file with minimal operator input, enabling the semi-automated extraction of data from articles into database load files. Preliminary testing of the tools shows promising performance in its ability to save time. The accuracy of an AI-generated load file is computed by comparing the presence and equality of extracted values to those in the respective human-created load file. These comparisons are made between dataframes containing the extracted data, so the score is robust to minor formatting differences within the files. Early testing has shown accuracies around 20%, but this low score is often not reflective of the often small effort needed to correct an AI-generated file. Future work will seek to expand the usage of this accuracy metric to robustly quantify the performance of these tools.

3.1 Computational Performance

This work was has used a Quadro RTX 6000 GPU for model finetuning and inference. The run duration of the data extraction tools increase with the number of objects and the length of the text. Inference using INDUS typically takes less than one minute per table, while prompts using AstroSage return responses in roughly 5-10 minutes per object. The slow completion speed of AstroSage prompts motivates the investigation of methods not based on decoder models to extract data from the text of an article (see S4.1) in less time, which will also aid in large-scale performance quantification.

4 Conclusions

This early work has shown that AI-powered tools, when combined with other programmatic methods and supervised by humans, can enhance the data ingestion pipelines at NED and NEA. While the results from early versions of these tools can suffer in accuracy, the time it takes to generate and correct a file can also be less than the time it would take to make the file by hand. Transformer-based AI is useful at several junctures of this work, but reliance on these methods alone were found to be insufficient for some subtasks of this project. Both automated and human verifications within the pipeline are needed due to the inaccuracy of AI-derived solutions (although there is potential for improvement). There are also practical limitations to the effectiveness and accuracy of automated data

extraction from the literature due to issues with the way data are sometimes published. Examples include: ambiguous object names, which are typically truncated coordinate-based names that cannot be accurately cross-identified using the NED and NEA name resolvers; data with missing uncertainties; omission of the reference frame for some measurements; and critical data linked to URLs that are no longer working. Many of these issues can be solved if authors and referees are more careful about following best practices for publishing data in the astronomical literature (Chen et al., 2022b). This work is in active development and will continue to be improved upon in the coming months.

4.1 Future Work

Prototype versions of these tools are being tested in production contexts. Operators have been asked to provide feedback which will be addressed to make improvements.

We will also seek to improve the automated training data labeling process for the object name detection model. Name validation tools will be used to eliminate false positive token labels and a broader search (i.e. searching each article for every name type) will reduce false negative token labels. This finetuned INDUS model and accompanying training data will be shared on the HuggingFace platform once its performance is improved.

Supervised finetuning of INDUS and AstroSage for the extraction of other data types will also be investigated, as decoder finetuning has been shown to increase the accuracy of related tasks (Zhao et al., 2024). This can be done at the document level for most data types, as the location of extracted data within an article is not retained by NED or NEA.

Additionally, data from external sources provided in links within articles will be accounted for where possible, as well as the units of numerical values (including automatic conversion). The evolving nature of the language used in astronomical journal article as new methods are employed or missions launched necessitates the periodic retraining of literature models. This will begin by replacing the old Standford/Doc2Vec-based relevance classification models with the encoder LLM INDUS, as discussed in S2.2. Other extensions, such as the consideration of images, will be approached in the future. While models adapted to the domain of astronomy have been shown to achieve better performance on astronomy-related tasks than models trained on broader contexts (e.g. Grezes et al., 2022, 2024; Bhattacharjee et al., 2024; de Haan et al., 2025), this work would benefit from a comparison between models like INDUS and AstroSage to modern frontier models from groups such as OpenAI and DeepSeek.

5 Acknowledgements

This work is supported by the NASA/IPAC Extragalactic Database (NED) and NASA/IPAC Exoplanet Archive (NEA), which are funded by the National Aeronautics and Space Administration under Award Number 80NSSC21M0037 and operated by the California Institute of Technology. We thank Douglas McElroy and Denis Zaytsev for important early contributions to this work.

References

Bishwaranjan Bhattacharjee, Aashka Trivedi, Masayasu Muraoka, Muthukumaran Ramasubramanian, Takuma Udagawa, Iksha Gurung, Nishan Pantha, Rong Zhang, Bharath Dandala, Rahul Ramachandran, Manil Maskey, Kaylin Bugbee, Mike Little, Elizabeth Fancher, Irina Gerasimov, Armin Mehrabian, Lauren Sanders, Sylvain Costes, Sergi Blanco-Cuaresma, and 17 others. 2024. INDUS: Effective and Efficient Language Models for Scientific Applications. *arXiv e-prints*, arXiv:2405.10725.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language Models are Few-Shot Learners. *arXiv e-prints*, arXiv:2005.14165.

Tracy X. Chen, Rick Ebert, Joseph M. Mazzarella, Cren Frayer, Scott Terek, Ben H. P. Chan, David Cook, Tak Lo, Marion Schmitz, and Xiuqin Wu. 2022a. Classification of Astrophysics Journal Articles with Machine Learning to Identify Data for NED. , 134(1031):014501.

Tracy X. Chen, Marion Schmitz, Joseph M. Mazzarella, Xiuqin Wu, Julian C. van Eyken, Alberto Accomazzi, Rachel L. Akeson, Mark Allen, Rachael Beaton, G. Bruce Berriman, Andrew W. Boyle, Marianne Brouty, Ben H. P. Chan, Jessie L. Christiansen, David R. Ciardi, David Cook, Raffaele D'Abrusco, Rick Ebert, Cren Frayer, and 26 others. 2022b. Best Practices for Data Publication in the Astronomical Literature. , 260(1):5.

Tijmen de Haan, Yuan-Sen Ting, Tirthankar Ghosal, Tuan Dung Nguyen, Alberto Accomazzi, Azton Wells, Nesar Ramachandra, Rui Pan, and Zechang Sun. 2025. Achieving GPT-4o level performance

in astronomy with a specialized 8B-parameter large language model. *Scientific Reports*, 15(1):13751.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv e-prints*, arXiv:1810.04805.

Tuan Dung Nguyen, Yuan-Sen Ting, Ioana Ciucă, Charlie O'Neill, Ze-Chang Sun, Maja Jabłońska, Sandor Kruk, Ernest Perkowski, Jack Miller, Jason Li, Josh Peek, Kartheik Iyer, Tomasz Różański, Pranav Khetarpal, Sharaf Zaman, David Brodrick, Sergio J. Rodríguez Méndez, Thang Bui, Alyssa Goodman, and 5 others. 2023. AstroLLaMA: Towards Specialized Foundation Models in Astronomy. *arXiv e-prints*, arXiv:2309.06126.

Jenny Rose Finkel, Trond Grenager, and Christopher D. Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, pages 363–370. Association for Computational Linguistics.

Jian Ge, Hui Zhang, Yongshuai Zhang, Yan Li, Dan Zhou, Haijiao Jiang, Pengjun Zhang, Xinyu Yao, Jiapeng Zhu, Yong Yu, Congcong Zhang, Zhenghong Tang, Jianqing Cai, Chaoyan Wang, Hongping Deng, Wen Chen, Kun Chen, Yingquan Yang, Xuliang Duan, and 50 others. 2024. Progress in the Earth 2.0 (ET) space mission. In *Space Telescopes and Instrumentation 2024: Optical, Infrared, and Millimeter Wave*, volume 13092, page 1309218. International Society for Optics and Photonics, SPIE.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 181 others. 2024. The Llama 3 Herd of Models. *arXiv e-prints*, arXiv:2407.21783.

F. Grezes, S. Blanco-Cuaresma, A. Accomazzi, M. J. Kurtz, G. Shapurian, E. Henneken, C. S. Grant, D. M. Thompson, R. Chyla, S. McDonald, T. W. Hostetler, M. R. Templeton, K. E. Lockhart, N. Martinovic, S. Chen, C. Tanner, and P. Protopapas. 2024. Building astroBERT, a Language Model for Astronomy & Astrophysics. In *Astromical Data Analysis Software and Systems XXXI*, volume 535 of *Astronomical Society of the Pacific Conference Series*, page 119.

Felix Grezes, Thomas Allen, Sergi Blanco-Cuaresma, Alberto Accomazzi, Michael J. Kurtz, Golnaz Shapurian, Edwin Henneken, Carolyn S. Grant, Donna M. Thompson, Timothy W. Hostetler, Matthew R. Templeton, Kelly E. Lockhart, Shinyi Chen, Jennifer Koch, Taylor Jacovich, and Pavlos Protopapas. 2022. Improving astroBERT using Semantic Textual Similarity. *arXiv e-prints*, arXiv:2212.00744.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. 2024. RULER: What's the Real Context Size of Your Long-Context Language Models? *arXiv e-prints*, arXiv:2404.06654.

Quoc V. Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. *arXiv e-prints*, arXiv:1405.4053.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv e-prints*, arXiv:1907.11692.

F. Matuszewski, N. Nettelmann, J. Cabrera, A. Börner, and H. Rauer. 2023. Estimating the number of planets that PLATO can detect. , 677:A133.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *arXiv e-prints*, arXiv:1301.3781.

Matthew T. Penny, B. Scott Gaudi, Eamonn Kerins, Nicholas J. Rattenbury, Shude Mao, Annie C. Robin, and Sebastiano Calchi Novati. 2019. Predictions of the WFIRST Microlensing Survey. I. Bound Planet Detection Rates. , 241(1):3.

Michael Perryman, Joel Hartman, Gáspár Á. Bakos, and Lennart Lindegren. 2014. Astrometric Exoplanet Detection with Gaia. , 797(1):14.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. LLaMA: Open and Efficient Foundation Language Models. *arXiv e-prints*, arXiv:2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023b. Llama 2: Open Foundation and Fine-Tuned Chat Models. *arXiv e-prints*, arXiv:2307.09288.

Wes McKinney. 2010. Data Structures for Statistical Computing in Python. In *Proceedings of the 9th Python in Science Conference*, pages 56 – 61.

Robert F. Wilson, Thomas Barclay, Brian P. Powell, Joshua Schlieder, Christina Hedges, Benjamin T. Montet, Elisa Quintana, Iain Mcdonald, Matthew T. Penny, Néstor Espinoza, and Eamonn Kerins. 2023. Transiting Exoplanet Yields for the Roman Galactic Bulge Time Domain Survey Predicted from Pixel-level Simulations. , 269(1):5.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *arXiv e-prints*, arXiv:1910.03771.

Hang Zhao, Qile P. Chen, Yijing Barry Zhang, and Gang Yang. 2024. Advancing Single and Multi-task Text Classification through Large Language Model Fine-tuning. *arXiv e-prints*, arXiv:2412.08587.