PROCEEDINGS OF SCIENCE

New ways to access PDG data

Juerg Beringer[†,*]

Lawrence Berkeley National Laboratory,
1 Cyclotron Road, Berkeley, CA 94720, USA

E-mail: jberinger@lbl.gov

In recent years, the data published by the Particle Data Group (PDG) in the *Review of Particle Physics* has primarily been accessed on the PDG web pages and in pdgLive, or downloaded in the form of PDF files. A new set of tools (PDG API) makes PDG data easily accessible in machine-readable format and includes a REST API that allows downloading of data in JSON format from pdgLive pages, a Python API, and downloadable database files containing the PDG data.
To find desired information, users either navigate to the corresponding review article or section in Particle Listings or Summary Tables on the web, use the new PDG API, or rely on a Google-based search of the PDG website. Large Language Models (LLM) combined with semantic search and Retrieval-Augmented Generation (RAG) are expected to allow enhancing the searching of PDG information in order to provide fine-grained accurate results.

We present the new PDG API, give examples of its use, and discuss a prototype implementation of a new PDG search tool.

42nd International Conference on High Energy Physics (ICHEP2024)
18-24 July 2024
Prague, Czech Republic

[*]Speaker
[†]On behalf of the PDG Collaboration

PoS(ICHEP2024)1023

1. Introduction

The Particle Data Group's (PDG) *Review of Particle Physics* provides a comprehensive summary of particle physics and related areas from cosmology and astrophysics. It is updated online each year and published in a scientific journal every two years. In 2,382 pages, the latest edition [1] gives in its "Summary Tables" world averages or best limits for about 10,000 quantities and provides curated and annotated lists of about 50,000 published measurements and limits in the "Particle Listings". The *Review of Particle Physics* also summarizes a wide range of topics in 120 review articles on constants, units, atomic and nuclear properties, the Standard Model and related topics, astrophysics and cosmology, experimental methods and colliders, mathematical tools, kinematics, cross-section formulae and plots, particle properties, and hypothetical particles and concepts.

The data and reviews published in the *Review of Particle Physics* are available online from the PDG website [2], the interactive pdgLive [3] for browsing particle data, and as the traditional printed "PDG Book" and "Particle Physics Booklet" (also available as web and Android versions for smartphones [4]) that served for decades as the primary means of distributing the *Review of Particle Physics*.

Various downloadable fixed-format data files have long made some of the PDG data, such as particle masses, widths, and PDG Monte-Carlo particle ID numbers, available in machine-readable format. With the 2024 edition of the *Review of Particle Physics*, a new PDG API (Application Programming Interface) is available that provides programmatic access in a modern format to all PDG data, including for the first time also branching fractions. This new PDG API is presented in the first part of this article.

The second part of this article discusses how recent AI tools, in particular Large Language Models (LLM), semantic search and Retrieval-Augmented Generation (RAG)[1] might be combined to enhance the searching of the *Review of Particle Physics* compared to what is currently available via commercial search engines such as Google.

2. Accessing PDG data in machine-readable format with the new PDG API

The new PDG API consists of three related tools aimed at different use cases, namely the interactive downloading of data in JSON format from pdgLive via a REST API, a Python API for high-level programmatic access to PDG data, and downloadable databases containing the PDG data in a single file. In the following, these tools will be briefly described. Comprehensive documentation is available [5].

2.1 PDG REST API

The PDG REST API allows the downloading of PDG data in JSON format via special URLs following the REST principles.[2] For example, the URL

```
https://pdgapi.lbl.gov/summaries/M061
```

[1] With RAG a LLM is used to summarize only the information provided in a request rather than drawing in general on its training data. This minimizes the danger of hallucinations, where a LLM generates made-up or irrelevant content.

[2] REST (Representational State Transfer) denotes conventions for exchanging data between programs. JSON (JavaScript Object Notation) is a widely used lightweight data-interchange format.

returns a JSON document with the data for the $D^*(2007)^0$ particle from the Summary Tables of the current edition of the *Review of Particle Physics*. In this URL, `/summaries` requests the data from the Summary Tables and `M061` is the PDG Identifier[3] for $D^*(2007)^0$. pdgLive uses this REST API to implement a button labelled "JSON" that allows the user to easily download in JSON format the data shown on a given page (see e.g. the $D^*(2007)^0$ pdgLive page at pdglive.lbl.gov/view/M061).

The REST API can be used both interactively in a browser and in scripts or programs. However, it is intended only for incidental use and access is rate-limited.

2.2 PDG Python API

While the REST API allows the incidental retrieval of PDG data in JSON format, the Python API provides a high-level interface for programmatic access to PDG data including navigation to and iteration over desired data. In many cases, this will be the most convenient and versatile way to access PDG data in machine-readable format. The PDG Python API is implemented in package `pdg` [6] and can be installed like any other Python package.

A sample use of the PDG Python API in a Jupyter notebook is shown in Figure 1, illustrating how to instantiate the API, access the information of a desired particle in different ways, and iterate over branching fractions.

The `pdg` package includes a copy of the PDG data that was current at the time the installed version of the package was released. By default this data is used, but the user can specify an alternative data source (see documentation [5]), for example one that contains the data from a different edition of the *Review of Particle Physics* or includes also historical data.

The Python API is designed to handle subtleties in PDG data in a user-friendly way by providing sensible default values expected to be correct in most cases, for example when the user is asking for a single "best value" of a quantity and there are several possible values whose choice depends on the specific use case. If such defaults are not desired, the API can be instantiated in so-called pedantic mode, where the user is warned with a Python exception in case of any ambiguities.

2.3 PDG database files

PDG database files are available from the PDG website and contain the data from a given edition of the *Review of Particle Physics* in the form of a single file in SQLite format.[4] Some versions of the database files also include historical values from the Summary Tables of previous editions.

Direct use of PDG database files is intended for cases where a software developer needs a local source of PDG data and the Python API is not suitable, for example in the case of a C++ application. These database files provide only low-level access to PDG data and a good understanding of how the data is organized, the meaning of internal flags, and the handling of special cases is required to use them. The Python API uses the database file of the current edition as its default data store.

[3] Because many quantities in particle physics are difficult to reference concisely and unambiguously via textual descriptions, PDG assigns each quantity a digital object identifier, the so-called PDG Identifier. These can be used across all PDG software to refer to a given quantity. PDG Identifiers are shown in pdgLive, and a complete list is available from the PDG website.

[4] SQLite provides a relational database in the form of a single file. It is widely used for local data stores and is available for many programming languages.

```
[1]: import pdg
     api = pdg.connect()
     print(api)
```

```
2024 Review of Particle Physics, data release 2024-05-31 02:00:13 PDT, API version 0.1.2
S. Navas et al. (Particle Data Group), Phys. Rev. D 110, 030001 (2024)
(C) Particle Data Group (PDG), data released under CC BY 4.0
For further information see https://pdg.lbl.gov/api
```

```
[2]: dstar0 = api.get_particle_by_mcid(423)
     # dstar0 = api.get_particle_by_name('D^*(2007)0')
     # dstar0 = api.get('M061')[0]
```

```
[3]: dstar0.description, dstar0.mcid, dstar0.mass, dstar0.is_meson, dstar0.quantum_I
```

```
[3]: ('D^*(2007)0', 423, 2.006852502141979, True, '1/2')
```

```
[4]: for bf in dstar0.branching_fractions():
         print(f'{bf.description:30}', bf.display_value_text)
```

```
D^*(2007)0 --> D0 pi0          (64.7+-0.9)%
D^*(2007)0 --> D0 gamma        (35.3+-0.9)%
D^*(2007)0 --> D0 e+ e-        (3.91+-0.33)E-3
D^*(2007)0 --> mu+ mu-         <2.5E-8
D^*(2007)0 --> e+ e-           <1.7E-6
```

Figure 1: Sample use of the PDG Python API in a Jupyter notebook. In cell [1], the PDG Python package `pdg` is imported, an instance `api` of the API is connected to the default PDG database that was installed with the package, and a summary is printed of what data will be used. Cell [2] demonstrates different ways of getting the data of the $D^*(2007)^0$ particle. In cell [3], different properties of the $D^*(2007)^0$ are retrieved and in cell [4] a table of its branching fractions is printed.

3. Enhanced searching of the *Review of Particle Physics*

When searching the *Review of Particle Physics* with the search box on the PDG website, which is currently powered by Google, one generally receives a list of links to PDF files, each potentially tens of pages long. While a line containing the search term will be shown for each file, this may not provide enough context to easily decide which of the files is most relevant and where among the many pages the desired information can be found. Thus the user has to open the different files and then search again within each file. Moreover, one can only search in English, which may not be the user's preferred language.

A more sophisticated PDG search tool might provide paragraph-level results and preview with relevance scores, fine-grained references with links that open on the relevant page, allow searching specifically for values from the Summary Tables or for tables or figures from review articles, and if desired provide a concise summary of the information found. Ideally all these features would be supported not only in English but equally in other languages a user might prefer.

Recent AI tools such as LLMs combined with semantic search and RAG allow building such a system with a relatively modest effort. To do this one can:

- Use a LLM to determine whether the user asks for the value of a specific quantity, for a figure or a table, or whether this is a general question about a topic covered in the *Review of Particle Physics*. By using a multi-lingual LLM, non-English queries can naturally be supported.

- Unless the user asks for the value of a specific quantity, perform a semantic search for matching paragraphs, tables or figures from review articles. For each match, a preview of the corresponding paragraph, table or figure is shown together with a link to the corresponding page in the review as well as a relevance score from the semantic search.
- For questions about specific values, use a LLM to determine matching quantities and corresponding PDG Identifiers. The PDG API is then used to retrieve the corresponding data and show a list of relevant quantities with their values.
- If desired, use a LLM to summarize the information found in the previous steps (i.e. use RAG to produce summaries).

Such a system fulfills the essential requirement that all results come exclusively from the *Review of Particle Physics*. Text, figures and tables come from PDG review articles. Specific data values are extracted from PDG Summary Tables and Particle Listings. By using RAG, even when requesting a summary of search results, the risk of hallucinations is minimized as much as possible.

The first challenge in building such a system comes from the fact that neither the PDF files of PDG review articles nor the corresponding LaTeX source files that use a PDG-specific style file can easily be converted by existing tools with 100% accuracy to a structured format such as Markdown that is suitable as the basis for semantic searching. For a first prototype implementation, Nougat [7] was used to convert all PDG reviews to Markdown, using Mathpix [8] to provide the necessary advanced support for equations and tables. In most cases this resulted in excellent high-quality conversions, but on occasion significant mistakes or conversion failures were noticed. Figures were extracted with PDFFigures2 [9], a semantic search was implemented using the Chroma vector database [10], and Langchain [11] was used as a standardized interface to different LLMs.

The resulting prototype is shown in Figure 2:

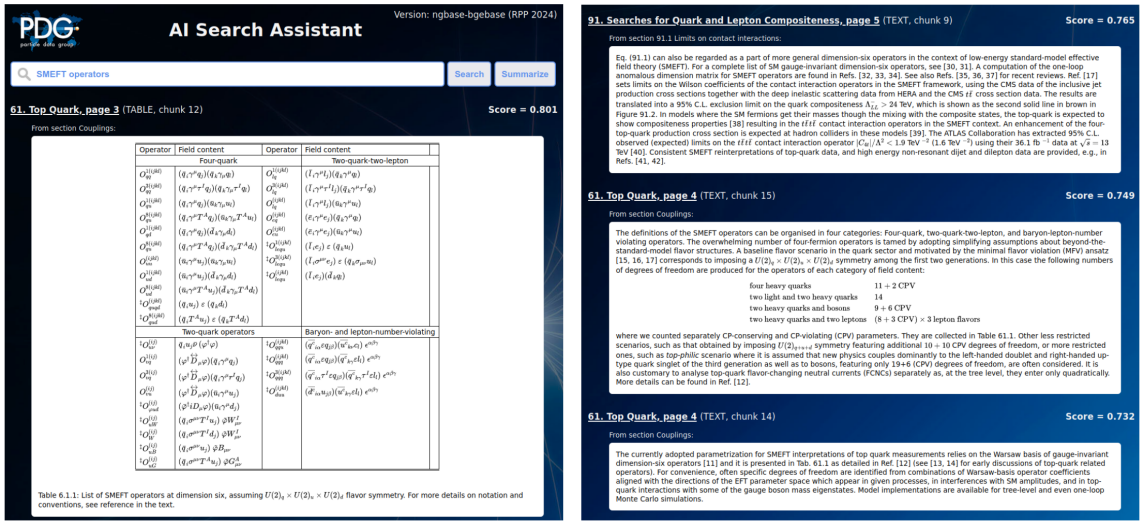

Figure 2: Screenshot of the prototype of an enhanced PDG search tool. For the sample search for "SMEFT operators", it shows the desired previews with fine-grained references to matching tables and paragraphs in PDG review articles as well as relevance scores and links to PDF files that open directly on the relevant pages.

4. Conclusions

With the 2024 edition of the *Review of Particle Physics*, a new PDG API with three tools aimed at different use cases provides programmatic access to PDG data, including for the first time also access to branching fraction data in machine-readable format. These tools include a Python API as the most versatile tool, direct download of data from pdgLive in JSON format via a REST API, and database files with the PDG data in SQLite format.

In order to enhance the searching of PDG data, the use of recent AI methods is explored, potentially allowing a more powerful searching of PDG data and reviews. Results from a first prototype have been encouraging and are now used to learn what matters and where improvements are needed. In particular, one of the hurdles is the requirement of a 100% accurate conversion of all PDG review articles into a suitable structured format such as Markdown.

Acknowledgements

This work was primarily supported by the Office of Science, Office of High Energy Physics of the U.S. Department of Energy under Contract No. DE–AC02–05CH11231, with important contributions from Japan (MEXT: Ministry of Education, Culture, Sports, Science and Technology) under the U.S.-Japan Science and Technology Cooperation Program in High Energy Physics, the Italian National Institute of Nuclear Physics (INFN), and the European Laboratory for Particle Physics (CERN). This research used the Lawrencium computational cluster resource provided by the IT Division at the Lawrence Berkeley National Laboratory. The PDG was designated as and is supported by the Office of Science as a Public Reusable Research (PuRe) Data Resource.

References

[1] S. Navas, *et al.* (Particle Data Group), *Review of Particle Physics*, Phys. Rev. D **110**, 030001 (2024)
[2] PDG website, pdg.lbl.gov
[3] pdgLive web application for browsing particle data, pdglive.lbl.gov
[4] Mobile Particle Physics Booklet, pdg.lbl.gov/booklet/
[5] Documentation of the PDG API, pdgapi.lbl.gov
[6] PDG Python API package `pdg`, pypi.org/project/pdg
[7] L. Blecher *et al.*, *Nougat: Neural Optical Understanding for Academic Documents*, arXiv:2308.13418
[8] Mathpix, mathpix-markdown-it
[9] C. Clark and S. Divvala, *PDFFigures 2.0: Mining Figures from Research Papers*, JCDL (2016), pdffigures2.allenai.org
[10] Chroma, www.trychroma.com
[11] LangChain, www.langchain.com