Scalable Knowledge-Graph Analytics at 136 Petaflops/s – Data Readme

Ramakrishnan Kannan, Piyush Sao, Hao Lu,
Drahomira Herrmannova, Robert Patton, Thomas Potok,
Vijay Thakkar, Richard Vuduc

August 12, 2020
v1.0

Acknowledgements

This manuscript has been authored by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan[1].

1 Introduction

This document describes how the datasets used in our paper "Scalable Knowledge-Graph Analytics at 136 Petaflops/s" [1] were prepared and can be used. We have used two datasets:

The COVID-19 dataset is based on the COVID-19 Open Research Dataset[2] (CORD-19) [3]. The CORD-19 dataset is a collection of scientific publications on SARS-COV-2, COVID-19, and other coronaviruses. The dataset is currently updated almost daily. In all processing and experiments described below we have used the version of the dataset from **June 30, 2020**.

The PubMed dataset is based on the latest complete release of PubMed[3] and Semantic MEDLINE Database[4] (SemMedDB) [2]. PubMed is a database of biomedical literature maintained by the National Library of Medicine (NLM).

[1] `http://energy.gov/downloads/doe-public-access-plan`
[2] `https://www.semanticscholar.org/cord19`
[3] `https://www.nlm.nih.gov/databases/download/pubmed_medline.html`
[4] `https://skr3.nlm.nih.gov/SemMedDB`

The largest collection of publications in PubMed is MEDLINE, a curated database of biomedical journals. SemMedDB [2] is a database of semantic predications – i.e. subject-predicate-object triples, that were extracted from the titles and abstracts of MEDLINE citations using SemRep [4], a tool developed by the NLM.

Both datasets were converted to a graph representation for processing by our DSNAPSHOT algorithm [1]. The format of the input and the output of DSNAPSHOT is described in Section 2 of this readme. All steps that were taken to produce the input format are described in Section 3.

2 Dataset description

This section describes the format of the input and output files of our DSNAPSHOT algorithm [1].

2.1 Input files

2.1.1 `graph.mtx`

The `graph.mtx` file contains the graph input for DSNAPSHOT. The matrix is stored in sparse coordinate format where each entry is composed of three values:

- the matrix row ID (i.e. the ID of a concept or a publication node),
- the column ID (again, the ID of a concept or a publication node),
- the value (edge weight between the two nodes).

The `graph.mtx` file was created using the `scipy.io.mmwrite`[5] function and ca be read using `scipy.io.mmread`[6].

The matrix describes the relations between biomedical concepts and the publications these concepts were extracted from. The first portion of the row and column IDs represents concepts (indices $[0; n_c - 1]$, where n_c is the total number of concept nodes), while the second portion represents papers (indices $[n_c; n_p - 1]$, where n_p is the total number of publication nodes). This is depicted in Figure 1. The meaning of the edge weights and how they were calculated is described in Section 3. The dataset provides two files that link the matrix row and column indices to unique concept and publication IDs: the `concepts.csv` and `papers.csv` files.

2.2 Output files

The DSNAPSHOT output is compose of the following two files.

[5] `https://docs.scipy.org/doc/scipy/reference/generated/scipy.io.mmwrite.html`
[6] `https://docs.scipy.org/doc/scipy/reference/generated/scipy.io.mmread.html`

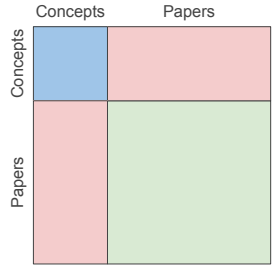

Figure 1: DSNAPSHOT input matrix. The first portion of the row and column indices represents concepts while the second portion of the indices represents publications. The bottom right quadrant of the matrix is constructed using citations between publications.

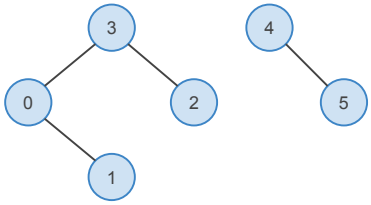

Figure 2: Example graph with two components.

2.2.1 `graph.dist`

The `graph.dist` file contains the distance between every pair of nodes in `graph.mtx`. The distances are stored in a dense matrix of size $[n_c+n_p; n_c+n_p]$. The row and column indices are organized in the same way as in the case of the `graph.mtx` matrix. In Python, the matrix can be loaded via `numpy.genfromtxt`[7].

2.2.2 `graph.aux`

The `graph.aux` contains information about the path between every pair of nodes in `graph.mtx`. The path information is stored in a dense matrix of size $[n_c + n_p; n_c + n_p]$. The row and column indices are organized in the same way as in the case of the `graph.mtx` matrix. Same as in the case of the `graph.dist` matrix, the `graph.aux` matrix can be loaded using `numpy.genfromtxt`.

The values in the matrix are of two types: a non negative integer which represents a concept or a paper ID, and the value -1 which means the two nodes in the graph do not have a path between them. Consider the example graph in Figure 2. The `graph.aux` matrix associated with this graph is shown in Table 1.

[7] `https://numpy.org/doc/stable/reference/generated/numpy.genfromtxt.html`

Table 1: Path matrix for graph in Figure 2.

0	0	1	3	3	-1	-1
1	0	1	0	0	-1	-1
2	3	3	2	3	-1	-1
3	0	0	2	3	-1	-1
4	-1	-1	-1	-1	4	5
5	-1	-1	-1	-1	4	5

Table 2: Two example entries from the `concepts.csv` file. Both concepts are represented by their UMLS IDs.

UMLS ID	matrix index
C0206419	0
C0030705	1

2.3 Supplementary files

2.3.1 `concepts.csv`

The `concepts.csv` file contains the mapping between indices in the `graph.mtx` matrix file and biomedical concepts. Concepts are represented by their Unified Medical Language System[8] (UMLS) IDs. The CSV file contains two columns: the first column contains the UMLS ID and the second column contains the index into the `graph.mtx` matrix for each concept. Table 2 shows two example entries from this CSV.

2.3.2 `papers.csv`

The `papers.csv` file contains the mapping between indices in the `graph.mtx` matrix file and PubMed papers. Papers are represented by their CORD-19 publication IDs or PubMed IDs. Same as in the case of the `concepts.csv` file, the `papers.csv` file contains two columns: the first column contains either the CORD-19 dataset index or the PubMed ID, and the second column contains the index into the `graph.mtx` matrix for each concept. Table 3 shows two example entries from this CSV.

3 Re-creating datasets used in our experiments

To create the `graph.mtx` file for both datasets, the COVID-19 dataset and the PubMed dataset, we did the following. The graph in both cases is constructed from SemRep [4] output. NLM has processed and shared both datasets and the

[8] `https://www.nlm.nih.gov/research/umls/index.html`

Table 3: Two example entries from the `papers.csv` file. The first paper is represented by its PubMed ID (11515789) and the second paper is represented by its CORD-19 ID (vhqafxi7).

PubMed/CORD-19 ID	matrix index
`11515789`	0
`vhqafxi7`	1

SemRep output of both is provided online. In the case of the PubMed dataset, we used the latest version of the Semantic MEDLINE database[9] [2] (version `semmedVER40_R` as of July 2020, the database is shared in SQL and contains SemRep output for all abstracts in MEDLINE). In the case of the COVID-19 dataset, we used the processed version of the CORD-19 database[10] [3] (version from June 30, 2020, the CORD-19 SemRep output is shared using the SemRep full-fielded output format[11]).

In both cases, we used the extracted predicates (entity-relation-entity triples, `PREDICATION` table in the Semantic MEDLINE database, `relation` type in SemRep output) to construct the concept-concept part of the matrix in Figure 1 (top left portion of the matrix). We counted the number of times each pair of concepts appears in a predicate together, regardless of the relation between them. The counts were converted to weights using the Jaccard similarity score:

$$w_{c_x c_y} = -\log \frac{|C_x \cap C_y|}{|C_x \cup C_y|}$$

Here C_x is the set of predicates containing concept x. Because Jaccard similarity score ranges from 0 to 1, where 1 represents perfect overlap between the predicates of two concepts, we apply $-\log$ to the similarity score to get a weight where lower values represent more similar concepts and vice versa.

Next, we used the list of extracted entities (`ENTITY` table in the Semantic MEDLINE database, `entity` type in SemRep output) to construct the concept-paper part of the matrix in Figure 1 (top right and bottom left portion of the matrix). We counted a number of times each concepts appeared in a paper and converted these counts to weights using the following formula:

$$w_{cp} = -\log \frac{P_{cp}}{P_p}$$

Here P_c is the number of times concept c appears in paper p and P_p is the total number of concepts in paper p. We again apply $-\log$ to the score to get weights where lower values represent concepts which are more important to a given paper and higher values are less important.

[9] `https://skr3.nlm.nih.gov/SemMedDB/download/download.html`
[10] `https://ii.nlm.nih.gov/SemRep_SemMedDB_SKR/COVID-19/index.shtml`
[11] `https://semrep.nlm.nih.gov/SemRep.v1.8_full_fielded_output.html`

To fill the final portion of the matrix in Figure 1 (bottom right portion), we used the PubMed dataset[12] to obtain information about citations between publications. In the case of the COVID-19 dataset, we only obtained this information for papers with a PubMed ID. For example, if paper with PubMed ID 1 cited a paper with PubMed ID 2, there would be a connection between these two papers in the graph. To convert these values to a weight, we used the following formula:

$$w_{p_x p_y} = -\log \frac{1}{N_{p_x} + N_{p_y}}$$

Here N_p represents the total number of citation relations of p (i.e. the total number of times p cited another paper or was cited by another paper). Again, we use $-\log$ to achieve weights where lower values represent higher importance.

3.1 Working with other publication sets

The COVID-19 dataset and the PubMed dataset are both based on sets of publications that were already processed with SemRep and are provided to the research community. When working with other publication sets an additional step that is required to convert the data into the format required by DSNAPSHOT is to process these publications using SemRep. We found the easiest way to do that is to use the online version of SemRep in batch mode[13]. This version of SemRep is hosted on NLM servers and can be used to process publications without having to install SemRep locally. For instructions on how to install and use SemRep please visit the NLM website[14].

3.2 Data filtering

For producing the COVID-19 `graph.mtx` file, we applied the following two concept filters. UMLS concepts are assigned one or multiple "semantic types" which are organized hierarchically into 14 major groups (the mapping is available on the NLM website[15]). We filtered out concepts based on their semantic types. Specifically, we filtered out 54 semantic types, for example "Daily or Recreational Activity" (`T056`), "Geographic Area" (`T083`), and "Educational Activity" (`T065`). We removed those concepts that were assigned only types from the list of the 54 types to remove – that is, if a concept was assigned additional types not on the list, we kept that concept in. The complete list of semantic types that were filtered out is provided in `removed_semtypes.txt`.

Additionally, we filtered out all concepts that represent common English words using the dictionary provided in the `pyEnchant`[16] library. In total, 5,331 concepts were removed using this filter. This filter removes concepts belong

[12] `https://www.nlm.nih.gov/databases/download/pubmed_medline.html`
[13] `https://ii.nlm.nih.gov/Batch/UTS_Required/semrep.shtml`
[14] `https://semrep.nlm.nih.gov/`
[15] `https://metamap.nlm.nih.gov/SemanticTypesAndGroups.shtml`
[16] `https://pypi.org/project/pyenchant/`

to semantic types such as "finding" (e.g. concepts "compliance" and "simple variant") and "qualitative concept" (e.g. concepts "normal color" and "highest grade") which might not be useful for understanding relations among more technical concept types such as "virus" and "molecular function". The list of words we used as stopwords are provided in the dataset in file `stopwords.txt`.

References

[1] Ramakrishnan Kannan, Piyush Sao, Hao Lu, Drahomira Herrmannova, Robert Patton, Thomas Potok, Vijay Thakkar, and Richard Vuduc. Scalable knowledge-graph analytics at 136 petaflops/s. In *Proceedings of the ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis (IEEE Supercomputing)*, 2020.

[2] Halil Kilicoglu, Dongwook Shin, Marcelo Fiszman, Graciela Rosemblat, and Thomas C Rindflesch. Semmeddb: a pubmed-scale repository of biomedical semantic predications. *Bioinformatics*, 28(23):3158–3160, 2012.

[3] Office of Science and Technology Policy. Call to action to the tech community on new machine readable COVID-19 dataset. Online, Mar 2020. Accessed: 2020-04-18.

[4] Thomas C Rindflesch and Marcelo Fiszman. The interaction of domain knowledge and linguistic structure in natural language processing: interpreting hypernymic propositions in biomedical text. *Journal of biomedical informatics*, 36(6):462–477, 2003.