Research and Development of Joint Language and Graph Embeddings for GES-DISC Search Engine

Xavier Evans
610.2, Armin Mehrabian
Greenbelt

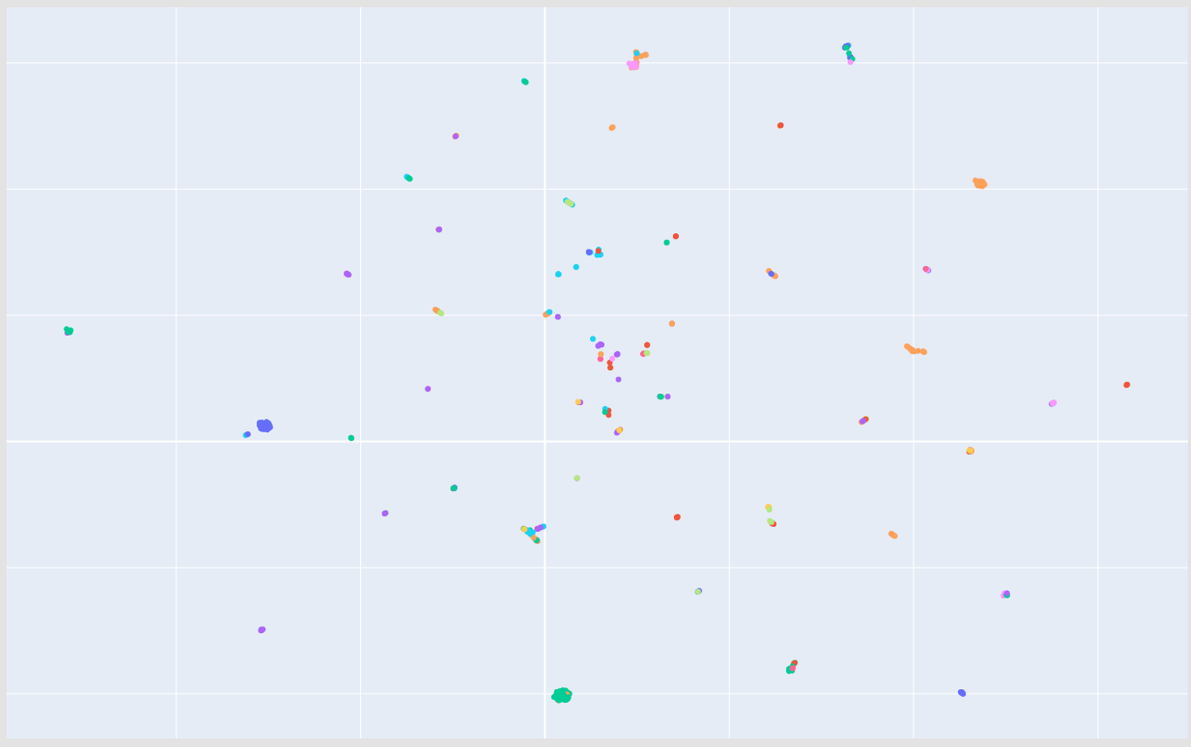
Introduction

The GES-DISC NLP search engine has been an invaluable resource for connecting scientists to the data sets they need for their work. Up until now, however, the search engine has only utilized natural language data to make recommendations to users. This is an intuitive starting point, yet by limiting ourselves to only using natural language data, we are neglecting the piece of meta-data which is, perhaps, the most essential to understanding how the data sets relate to each other: science keywords. By creating an 8-partite hierarchical graph of keywords and datasets, we can not only connect datasets to their keywords but also connect keywords to other keywords. This complex structure makes clear how all the datasets are related to each other, regardless of whether that relationship is close or distant. Embedding this structure and incorporating the resulting vector representations in the relevance scoring process can aid in providing users with more relevant results.

Each data set has one or more science keywords, and some of these are common among multiple data sets. Naturally, we would imagine that data sets with similar sets of keywords would be more similar and, thus, should be recommended to the user in the same instances. Yet, if we naively connect datasets to keywords to create a bipartite graph, similarities will only be detected when data sets' keywords are identical; related but not identical keywords will not be considered related at all unless we structure the keywords in a k-partite structure with $k > 2$. By incorporating the relationships between science keywords into a graph structure, we no longer neglect these indirect yet meaningful relationships which influence the recommendations.

Results

The following figure is a visualization of a UMAP two-dimensional dimensionality reduction from our original embeddings. Different colors correspond to different terms. If the embedding is effective, we expect similar data sets to be embedded close to each other. Data sets with similar keywords are similar, so we would hope to see clusters of the same color (clusters of datasets with keywords with common superfields); this would show that data sets that have related keywords are, in fact, reflected as being similar in this graph. Consider the following figure (zoomed examples follow the conclusion).



There are clusters of data sets that share terms in the same locations. This indicates that data sets that we would expect to be similar are, in fact, considered to be similar by the embeddings that our model has learned. This is a promising result, and these graph embeddings can be very helpful in recommending similar data sets to users.

Methodology

The graph devised to represent the relationship between datasets available in the GES-DISC search engine is 8-partite in nature and has heterogeneous nodes. There are seven node types corresponding to the seven types of keywords and one node type for datasets. The hierarchy of keywords from the most general to the most specific is as follows.

Category → Topic → Term → VariableLevel1 → VariableLevel2 → VariableLevel3 → DetailedVariable*

The partitions of the 8-partite graph can contain certain types of nodes as detailed below.

1. Category
2. Topic
3. Term
4. VariableLevel1, VariableLevel2, DetailedVariable, *Dataset*
5. VariableLevel2, VariableLevel3, DetailedVariable, *Dataset*
6. VariableLevel3, DetailedVariable, *Dataset*
7. DetailedVariable, *Dataset*
8. Dataset

This graph structure is far from homogeneous, but we can simplify it by neglecting the distinctions between keyword classes; in doing so, we lose potentially valuable information about how the type of the keyword interfaces with the structure of the graph. Yet, these minuscule distinctions likely have an effect negligible enough to be outweighed by the increase in intuitiveness and performance afforded by a more homogeneous graph. Then, there are two node types: keyword and dataset. To create a truly homogeneous graph, we treat keyword nodes as dataset nodes that function as an agglomeration of all descendent dataset nodes. This is the only way to determine node features for the keyword nodes which, themselves, have no substantial intrinsic natural language data on which to base the features; by agglomerating, we only need to rely on the features of descendent datasets and recursively do this from the leaves of the 8-partite hierarchical graph to the root keyword 'Earth Science.'

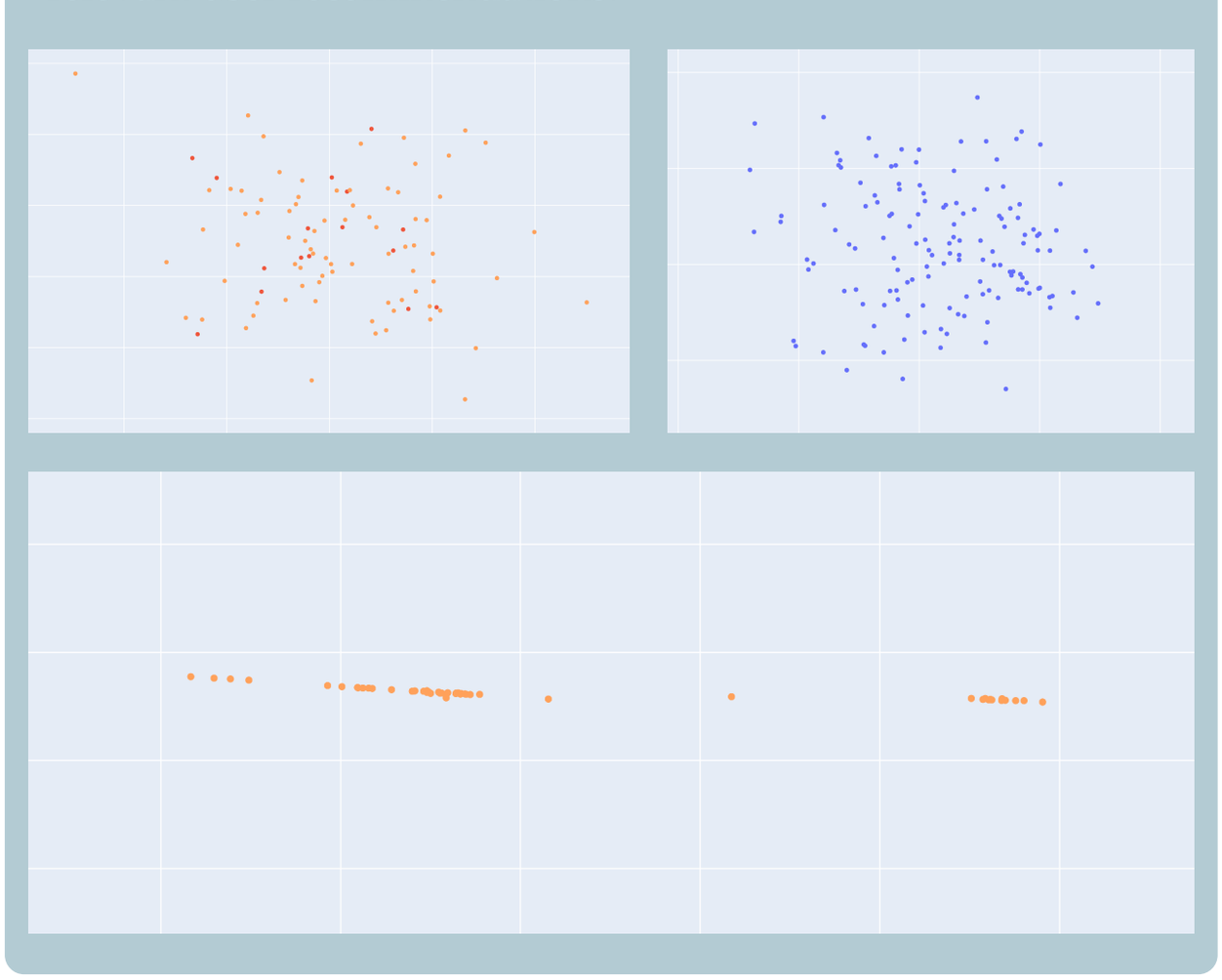
We fix dataset node features as the tf–idf scores for all words in the vocabulary of all dataset abstracts for each dataset. As there are 6399 tokens in this vocabulary, each dataset has 6399-dimensional node features. Each keyword node's features are calculated as the average of the tf–idf scores of all descendent datasets. This circumvents the issue of keyword nodes lacking natural language data, and it is reasonable given that we would expect a keyword node's features to be the exact middle ground of all its descendent datasets. Considering the keyword 'Earth Science,' since it is the root of the graph, it has dominion over the natural language data for all datasets; the tf–idf scores should, thus, be calculated from all this data, and taking the average achieves this. With keyword nodes that have cleverly-determined node features that stand in congruence to those of dataset nodes, we now have a homogeneous graph that can embedded using Stanford's GraphSAGE and the PyTorch Geometric Python library.

** DetailedVariables can be subfields of not only VariableLevel3s but also Terms–VariableLevel2s. It was only placed last for neatness.*

Conclusion

By only relying on natural language data, the GES-DISC search engine was not capitalizing on all available information. There is rich information to be extracted from the highly structured nature of data sets and their keywords. By representing the structure as a graph and applying machine learning techniques to this graph, we can better learn how these data sets relate to each other, including more distantly related ones (an impossible task before).

These insights can be used in tandem with the natural language processing techniques already in use. We use natural language processing to place the user's query in the embedding space. Then, we can easily recommend the closest datasets using vector distance metrics, leveraging the best that natural language and graph machine learning have to offer. This model can prove more effective for producing more relevant user recommendations.



Acknowledgments

Cohen, E. (2018, April 23). Node2vec: Embeddings for graph data. Medium. Retrieved August 2, 2022, from https://towardsdatascience.com/node2vec-embeddings-for-graph-data-32a866340fef

Hamilton, W. L., Ying, R., & Leskovec, J. (2017). Inductive Representation Learning on Large Graphs. arXiv. https://doi.org/10.48550/ARXIV.1706.02216

Long, Bo & Wu, Xiaoyun & Zhang, Zhongfei & Yu, Philip. (2006). Unsupervised learning on K-partite graphs. Proc. SIGKDD 2006. 317-326. 10.1145/1150402.1150439.

Maklin, C. (2019, July 21). TF IDF | TFIDF Python example. Medium. Retrieved July 28, 2022, from https://towardsdatascience.com/natural-language-processing-feature-engineering-using-tf-idf-e8b9d00e7e76