Exceptional service in the national interest

Sandia National Laboratories

Sequence Embeddings using LSTM Networks: Applications to Cybersecurity

Apurva Gandhi, Mahimna Kelkar, Joey Ingram, Shawn Martin
{agandhi, mkelkar, jbingra, smartin}@sandia.gov

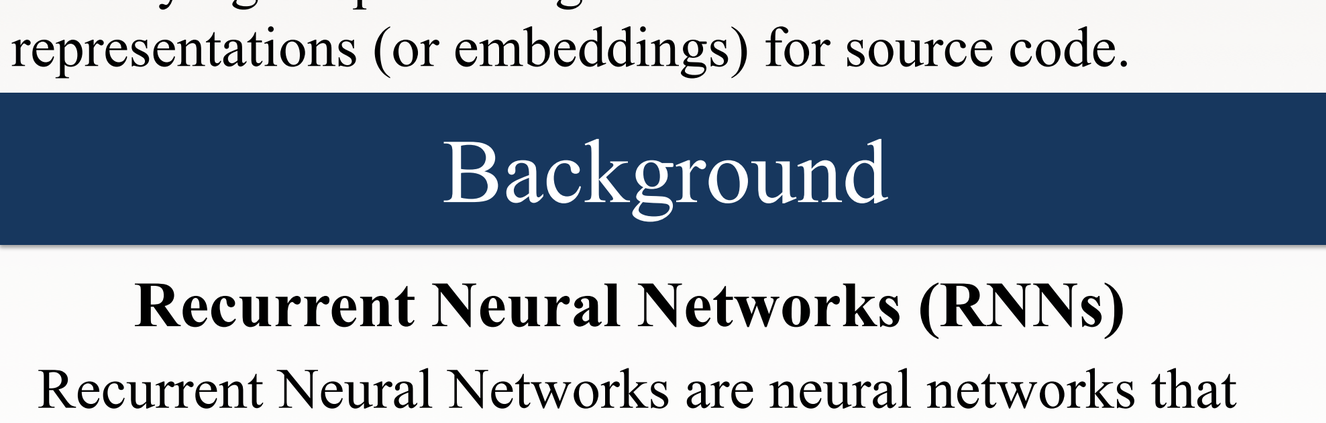
Abstract

With the increasing number of source-code-based cyber threats, analysis of source code is highly critical to cyber security. Automating this analysis would reduce costs and increase efficiency of threat mitigation. In this project, we explore three different approaches for automating source code analysis using deep learning, while progressively enhancing practical implications. We start with supervised classification and clustering of code, transition to supervised translation of pseudocode to code, and conclude with unsupervised clustering of source code. For each approach, we use an LSTM as our underlying deep learning model to create vector representations (or embeddings) for source code.

Background

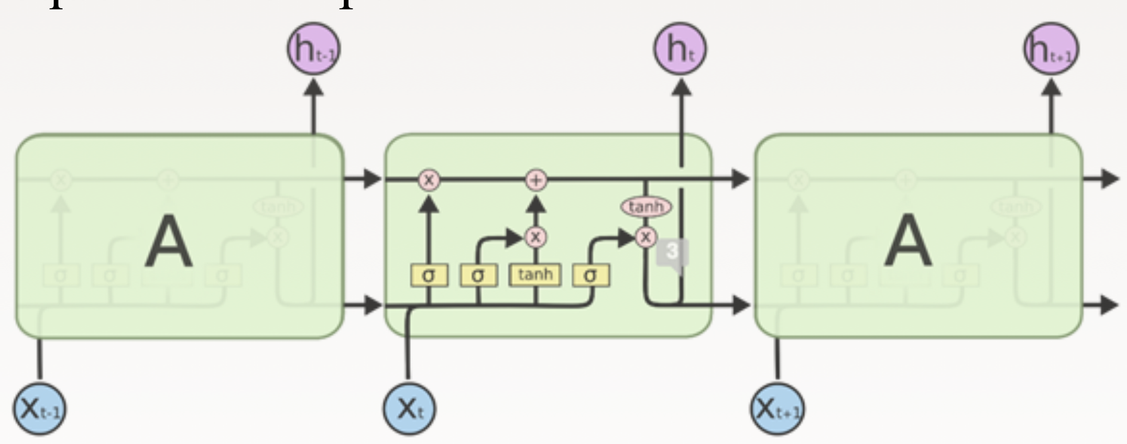
Recurrent Neural Networks (RNNs)

Recurrent Neural Networks are neural networks that are able to take in sequential input by reading in an element of a sequence at different timesteps and maintaining a state of previous timesteps. In our project, we model source code as sequential data. RNNs are an extremely flexible architecture, as they can also output sequential data if desired. This sequence to sequence architecture is at the heart of many neural machine translation algorithms.



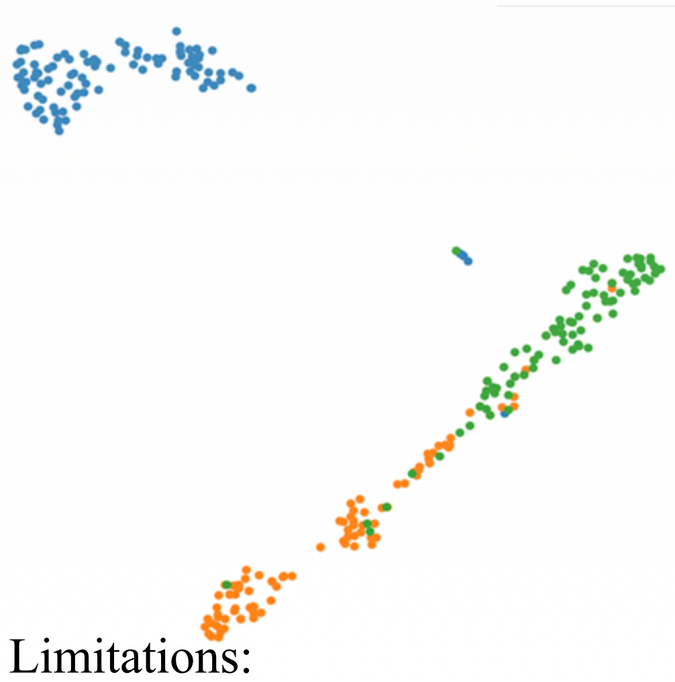
Long Short-Term Memory Networks (LSTMs)

While in theory vanilla RNNs should be able to learn long term dependencies in a sequence, in practice they often do not. A Long short-term memory Network (LSTM) is a special type of RNN with an architecture tailored to be able to handle long term dependencies. It is used in many of state-of-the-art deep learning applications that have sequences as inputs.



Methods

Supervised Code Clustering



Clustering of labelled data taken from the algorithm website GeeksForGeeks [1]

Sorting
Searching
Dynamic Programming

Limitations:

- Good labelled datasets for source code are hard to find.
- It may not possible to know beforehand all possible clusters

U.S. DEPARTMENT OF ENERGY
National Nuclear Security Administration

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525. SAND #2018-7832 D

Exceptional service in the national interest

Sandia National Laboratories

Supervised Sequence to Sequence Translation: Generation of Code from Psuedocode

- To build a dataset for (pseudocode, code) pairs, we used Pseudogen [2], a tool based on statistical machine translation that generates pseudocode from Python code.
- Building an Encoder-Decoder LSTM model from this data, we solve the harder problem of generating code from pseudocode.

Correct Code Generation Examples

> define the get_many method with self class instance, keys and version set to None as arguments.
< def get_many(self, keys, version=None):

> if e.errno equals to errno.EEXIST,
< if e.errno == errno.EEXIST :

> increment weight by integer 2.
< weight += 2

> call the method router.db_for_write with argument self.cache_model_class, substitute the result for db.
< db = router.db_for_write(self.cache_model_class)

> do nothing
< pass

Legend:
> Input pseudocode line
< Output code line

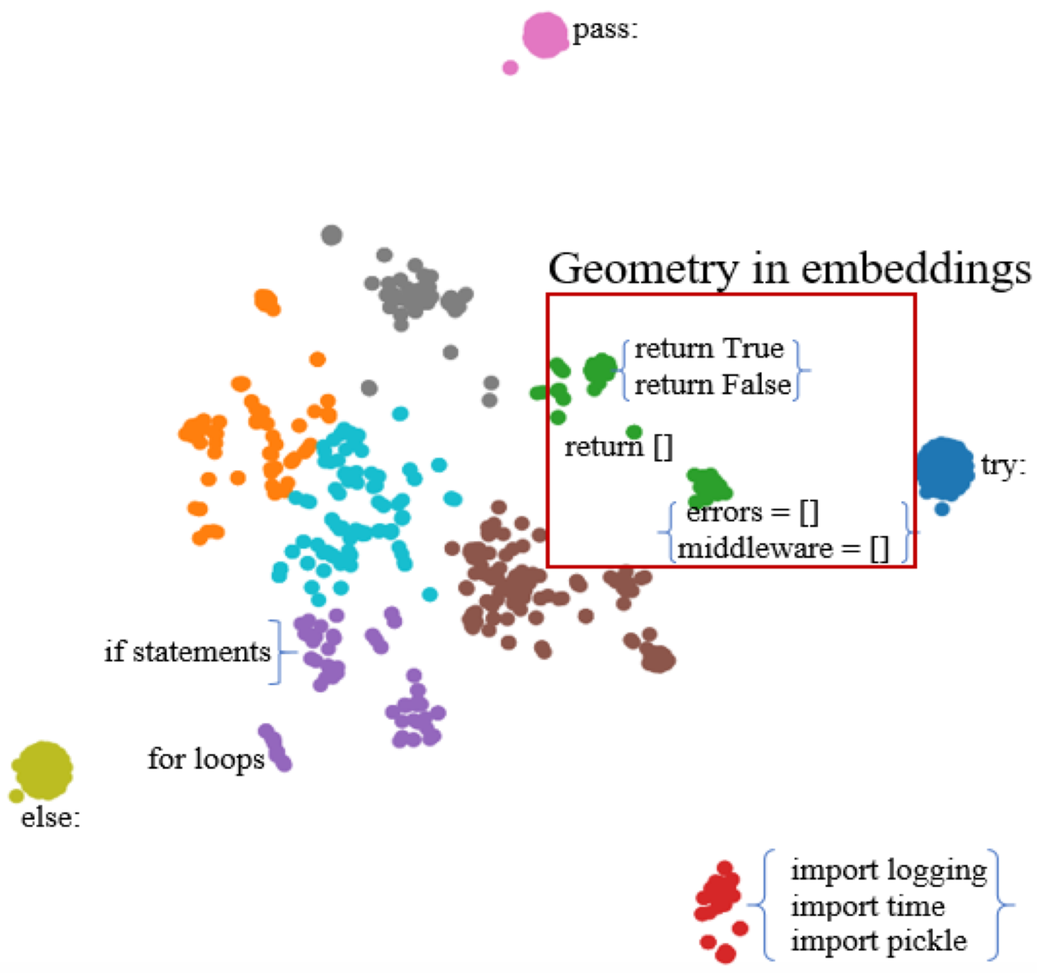
Incorrect Code Generation Examples

> if self has an attribute 'error_dict',
< if hasattr(self, name) :

> if self.dir path doesn't exists,
< if not os.path.exists (self.path) :

Unsupervised Code Clustering: A realization of underlying geometry

- Plotting vector embeddings from the supervised translation suggested the presence of a geometric relation between the vectors.
- Given the source code, without any labels, we used an LSTM autoencoder to recreate the input. The vector representations of the source code lines were obtained from the encoder LSTM.
- After dimensionality reduction using PCA and t-SNE, we plotted the vector embeddings.
- From the graph, we notice syntactic clustering of code along with the presence of some spatial structure as shown in red box.



Conclusion

- Recurrent deep learning models such as LSTMs demonstrate great ability in learning syntax-based vector representations of source code. This not only helps encode sequential data into vectors, but also maintains meaningful geometric relations amongst the representations.
- Applied to cyber security, such vector representations of code can be used to cluster source code in an unsupervised fashion for analysis of threats or even translate between languages to aid source code analysis.

Future Work

- Encoding functions/logical blocks rather than lines to obtain semantic geometric relations in code.
- The vector embeddings created by our work provide a general language agnostic representation of code, and so can be used to easily move source code between different languages or perform language agnostic source code analysis.

References

[1] www.geeksforgeeks.org
[2] Fudaba et. al. *Psuedogen: A Tool to Automatically Generate Pseudocode from Source Code*. ASE 2015

U.S. DEPARTMENT OF ENERGY
National Nuclear Security Administration

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525. SAND #2018-7832 D