Learning Code Embeddings from Abstract Syntax Graphs via Graph Attention Networks

Alexandru-Gabriel Sîrbu
Babeș-Bolyai University

WeADL 2025 Workshop

The workshop is organized under the umbrella of WinDMiL, project funded by CCCDI-UEFISCDI, project number PN-IV-P7-7.1-PED-2024-0121, within PNCDI IV

Overview

Problem statement

- Embeddings - dense vector representations of data
- Capture essential features, relationships, and contextual information
- Usage:
  - Transfer learning
  - Training data is limited

Word embeddings example

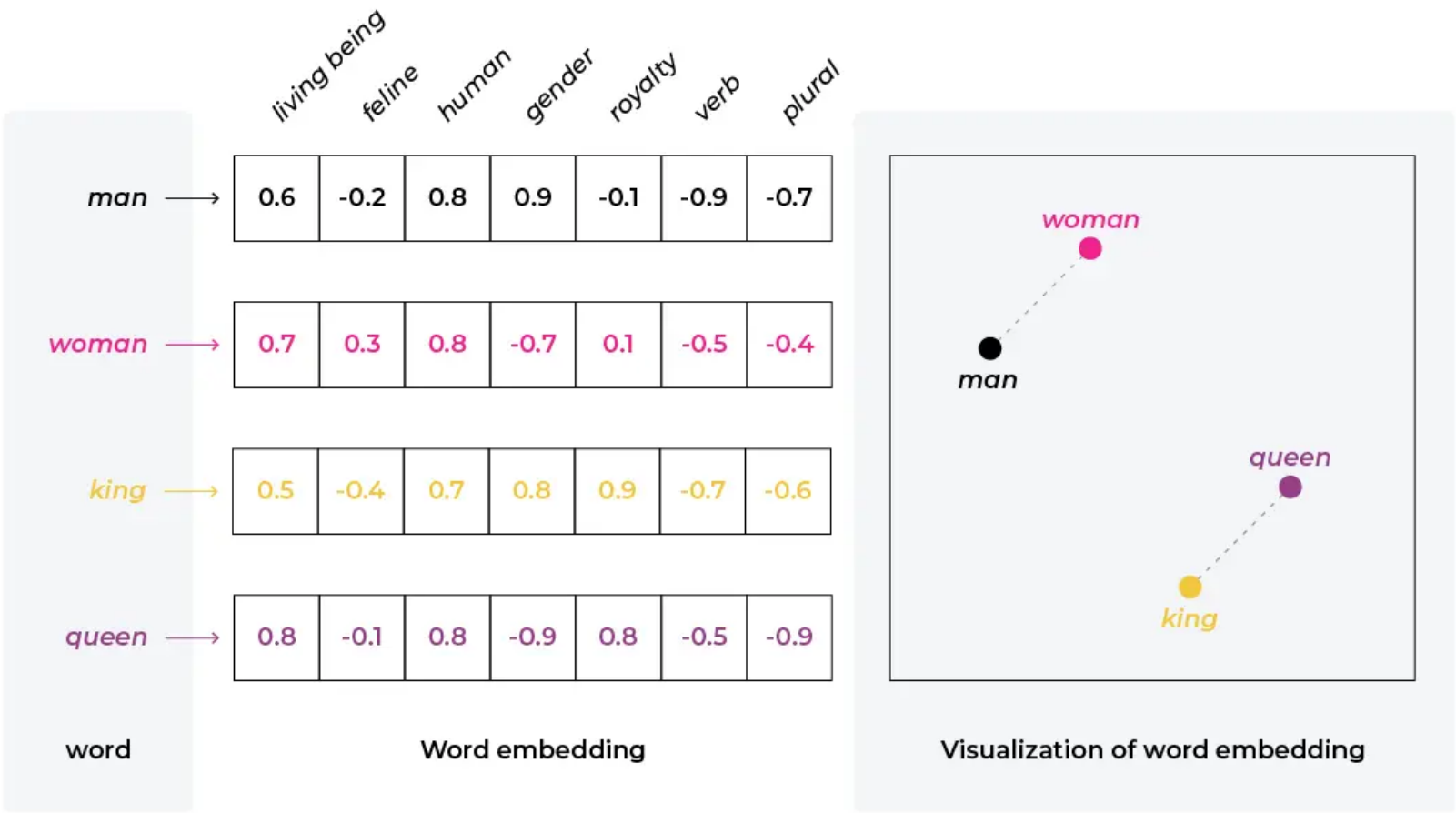

Figure: Example of word embeddings [Cas23]

Image embeddings example

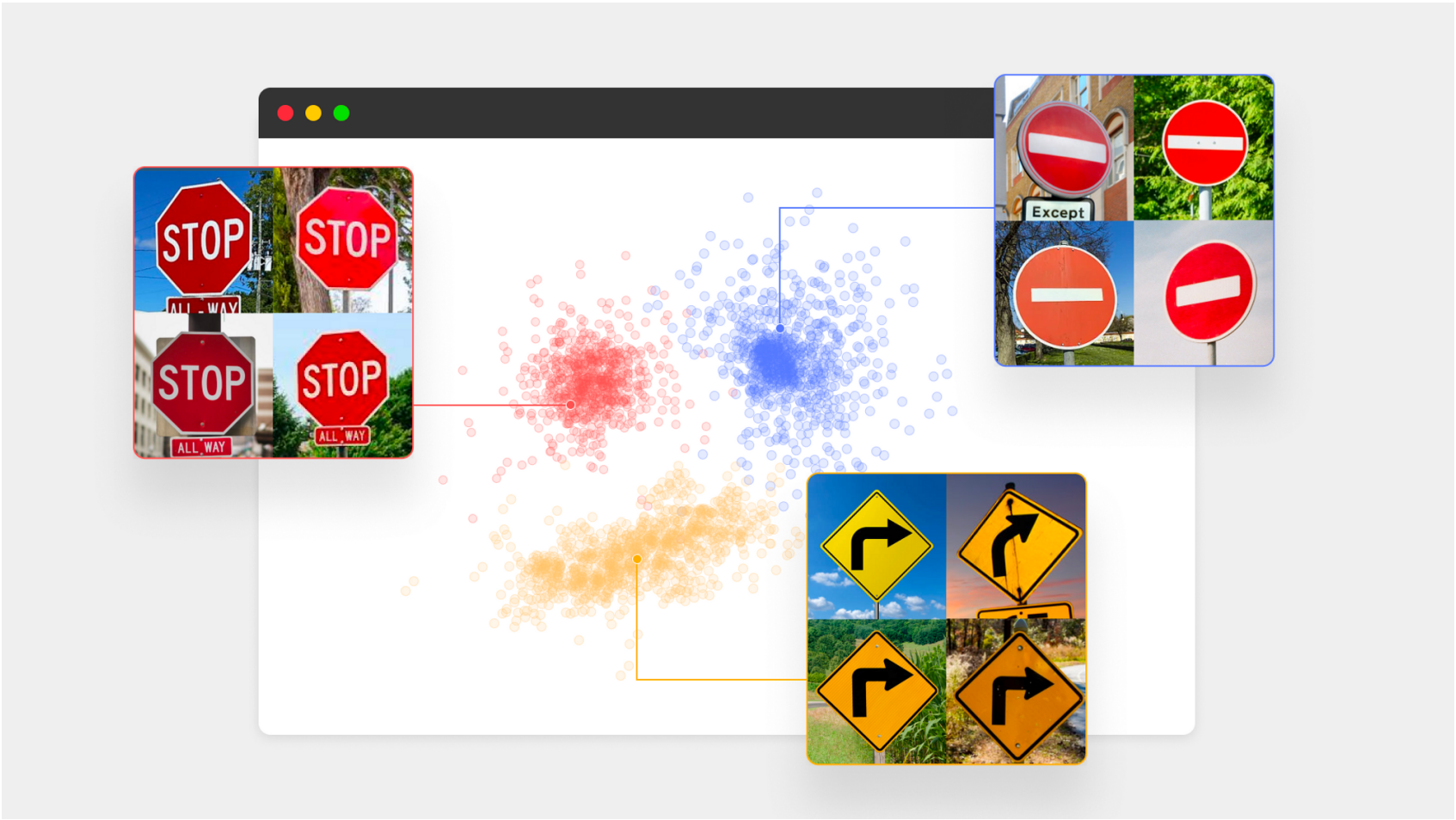

Figure: Example of image embeddings [Fat24]

Evaluation

- Incorporate in related tasks
- Clusterization of similar inputs

BERT [Dev18]

- Trained only on natural language text
- Masks 15% of the input tokens
- Predicts masks based on context
- Integrates next-sentence prediction

BERT Training example

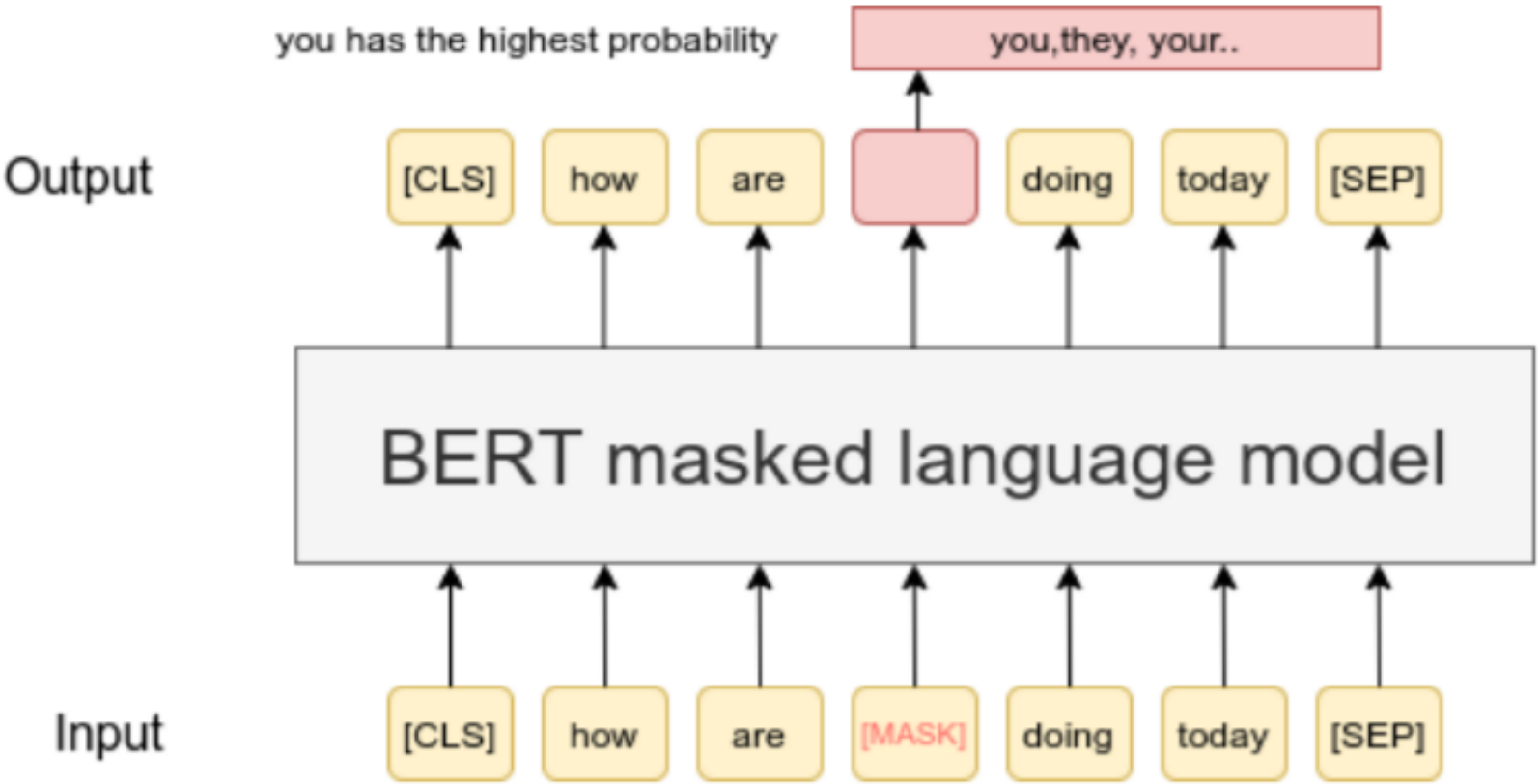

Figure: Example of BERT masked token prediction [Sha23]

CodeBERT [FGT+20]

- Based on BERT
- Trained on natural language text and programming languages
- Code snippets are treated as simple tokens
- Bimodal - can handle both natural language and programming language data

CodeT5 [WWJH21]

- Encoder-Decoder architecture
- Bimodal
- Optimized for code-related tasks
  - Denoise code
  - Mask tokens
  - Tag and predict masks

Embeddings evaluation for LLMs [PMCF25]

- Evaluated TF-IDF, BERT, Ada-002 from OpenAI, Falcon, LLaMA-2
- Algorithms: K-Means, K-Means++, Agglomerative Hierarchical Clustering, Fuzzy CM, Spectral Clustering
- Metrics: F1-score, Rand index, homogenity, Silhouette score, CHI score
  - Used both supervised and unsupervised metrics

BERT embeddings evaluation [YAH23]

- Task: group newspaper articles into their categories
- Algorithms: K-Means, DBSCAN
- Metrics: Silhouette score, Dunn index

Our proposal

- Literature approaches:
  - Treat code as simple tokens
  - Same model for text and code
- Encode the code to capture structure knowledge
- Single responsibility principle: embeddings just for code
- Evaluation on clusterization of similar data
- Compare to CodeBERT and CodeT5

Abstract Syntax Tree

- Hierarchical structure
- Describes relationships within the code
- Can be converted to/from code

Abstract Syntax Tree example

```
x = a + b;
y = a * b;
while (y > a)
{
    a = a + 1;
    x = a + b;
}
```

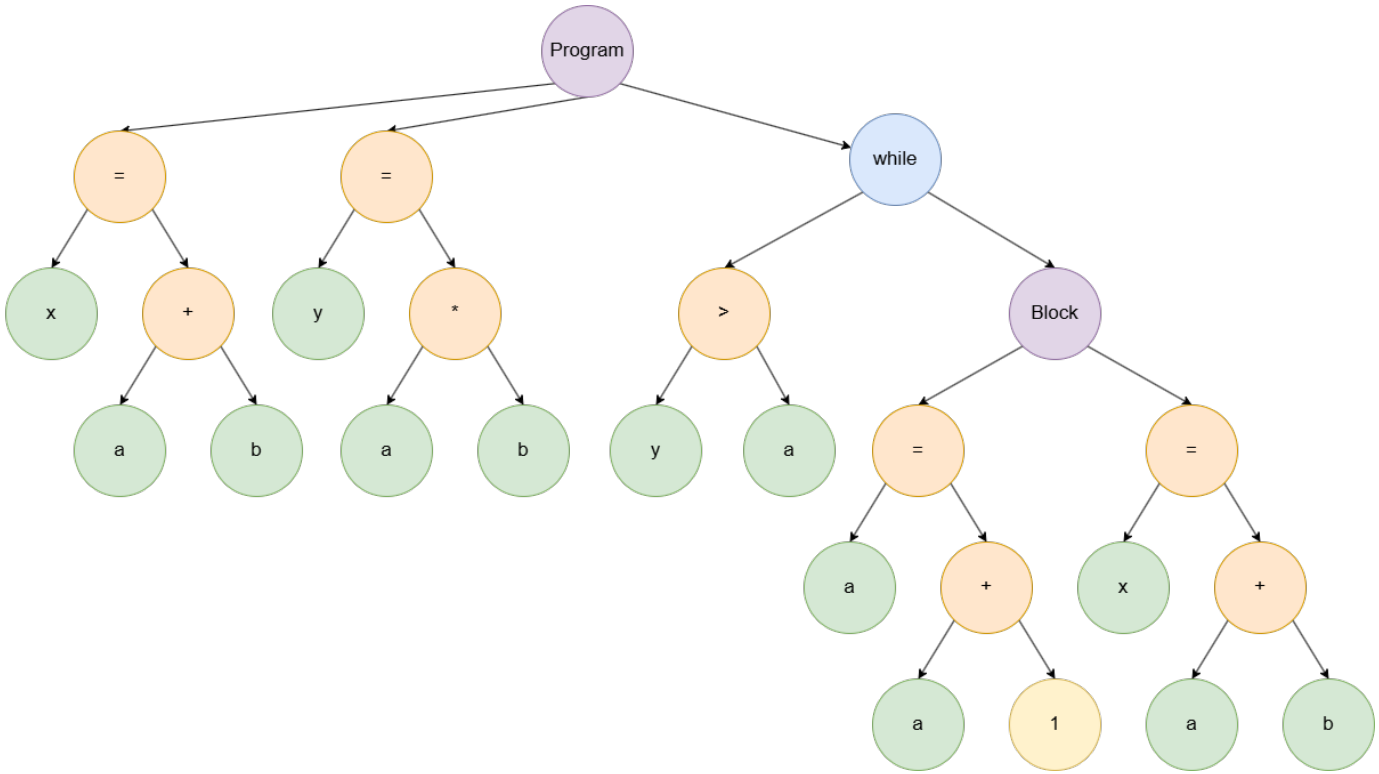

Figure: Comparison between a section of code (left) and its equivalent Abstract Syntax Tree (right)

Abstract Syntax Tree - issues

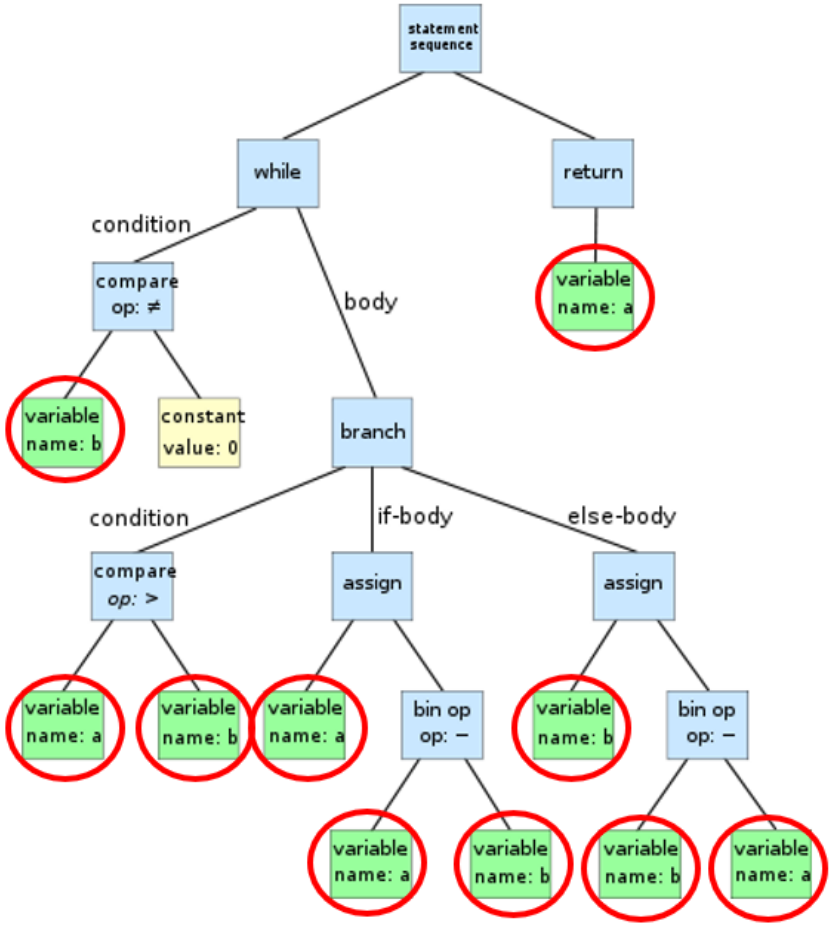

Figure: AST highlighting the redundancies in the leaves

Abstract Syntax Graph

- Compact alternative of an AST
- Repeated leaf nodes are concatenated
- Reduces the number of nodes generated
- Edges are inserted between two nodes of the same parent - introduces order
- Cycles are introduced into the structure - models re-usability

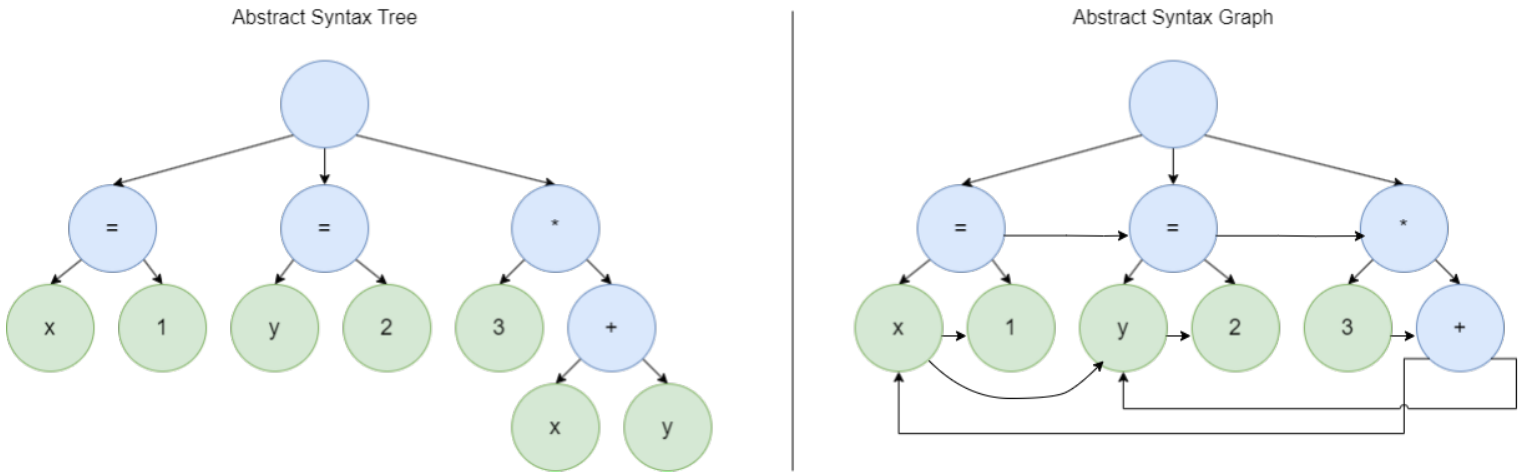

Figure: Comparison between an AST (left) and an ASG (right)

Methodology

- Encode the code as an ASG
- Train on:
  - Next node prediction
  - Next edges prediction
- Evaluate by clusterization

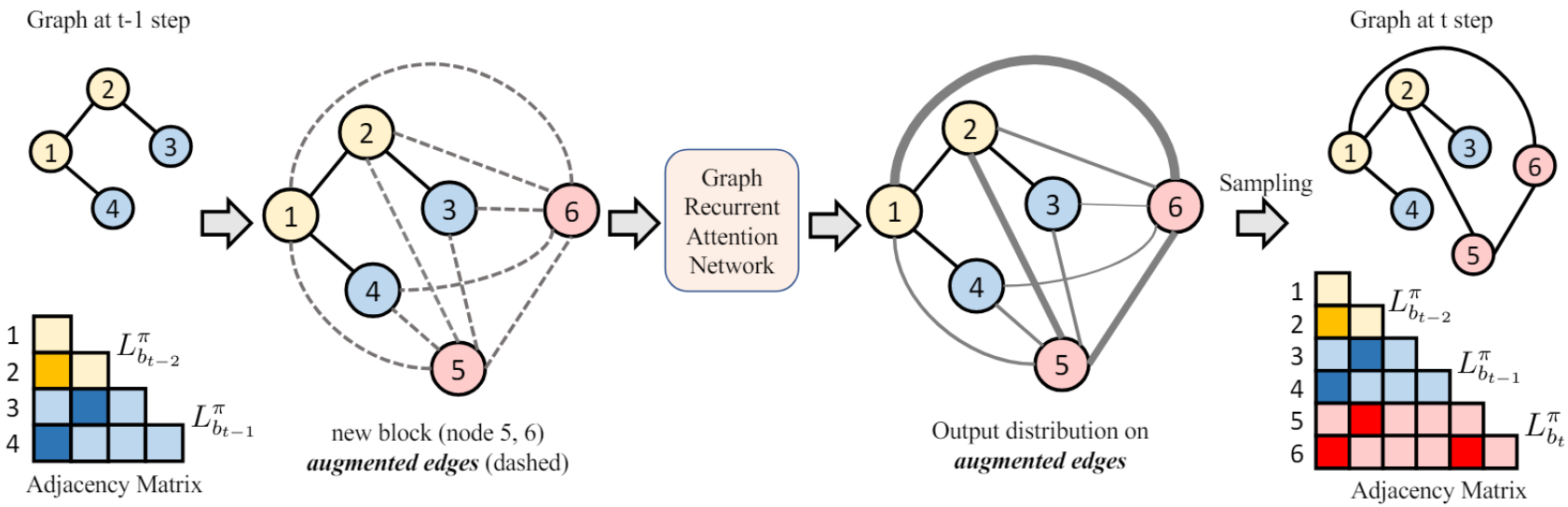

Figure: Overview of the graph generation model [LLS+19]

Model architecture

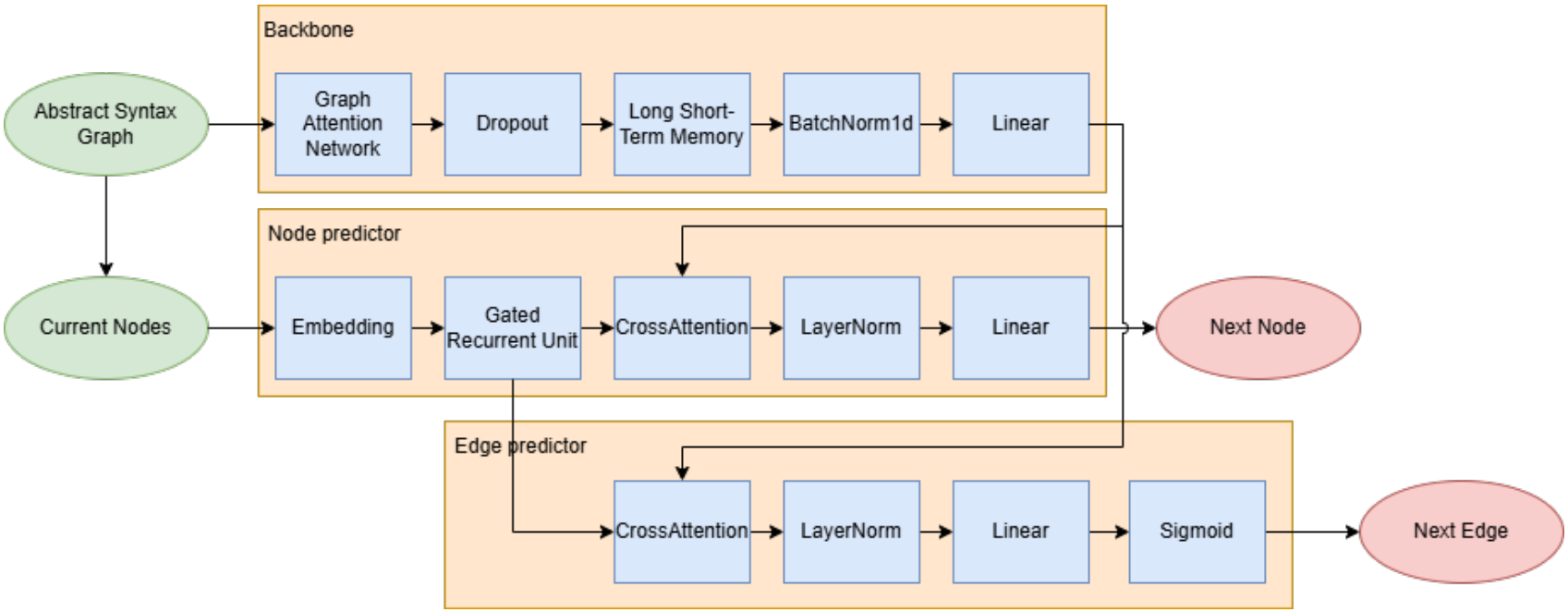

Figure: Abstract Syntax Graph-based model architecture

Clusterization

- Distance: Cosine Similarity
- Evaluation metric: Silhouette score
- Algorithms: K-Means, DBSCAN, Spectral Clustering

K-Means

- Partitions data into k clusters
- Data belongs in the cluster with the nearest mean
- Requires the use of elbow method, since k is not known beforehand

DBSCAN

- Density based clustering algorithm
- Groups together points that are closely packed
- Marks outliers
- Finds non-linearly separable clusters

Spectral Clustering

- Uses k eigenvalues of the similarity matrix of the data
- Groups data in lower dimensions
- Requires the use of elbow method for k

Results

Table: Silhouette Scores for Clustering Embeddings from the proposed model, CodeBERT and CodeT5

Model	K-Means	DBSCAN	Spectral Clustering
CodeBERT	0.05976184	0.19362077	0.2615287
CodeT5	0.06064902	0.19752871	0.2663004
Proposed Model	**0.06255425**	**0.20006323**	**0.2694121**

Discussion

- Code as simple tokens $\Rightarrow$ programming language independence
- ASTs and ASGs require a parser
- Parser's quality influences our model's quality
- ASG embeddings capture structural relationships

Conclusions

- Powerful framework for code representation
- Increased quality of the embeddings
- Trained for code generation tasks
  - Code completion
  - Code summarization
  - Code translation

Future work

- Evaluate in code-related tasks, e.g., Software Defect Prediction
- Measure performance gain against ASTs
- Train on a task of Masked Node Prediction and Edge validation

Thank you!

Bibliography


Francisco Castillo.
Embeddings: Meaning, examples and how to compute, Jun 2023.


Jacob Devlin.
Bert: Pre-training of deep bidirectional transformers for language understanding.
arXiv preprint arXiv:1810.04805, 2018.


Shaistha Fathima.
Understanding and utilizing embedding in ml: A brief overview, Jan 2024.


Zhangyin Feng, Daya Guo, Duyu Tang, Nan Duan, Xiaocheng Feng, Ming Gong, Linjun Shou, Bing Qin, Ting Liu, Daxin Jiang, et al.
Codebert: A pre-trained model for programming and natural languages.
arXiv preprint arXiv:2002.08155, 2020.


Renjie Liao, Yujia Li, Yang Song, Shenlong Wang, William L. Hamilton, David Duvenaud, Raquel Urtasun, and Richard S. Zemel.
Efficient Graph Generation with Graph Recurrent Attention Networks.
In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 4257–4267, 2019.


Alina Petukhova, João P Matos-Carvalho, and Nuno Fachada.
Text clustering with large language model embeddings.
International Journal of Cognitive Computing in Engineering, 6:100–108, 2025.

Rayyan Shaikh.

Mastering bert: A comprehensive guide from beginner to advanced in natural language processing..., Aug 2023.

Yue Wang, Weishi Wang, Shafiq Joty, and Steven CH Hoi.

Codet5: Identifier-aware unified pre-trained encoder-decoder models for code understanding and generation. *arXiv preprint arXiv:2109.00859*, 2021.

Sumona Yeasmin, Nazia Afrin, and Mohammad Rezwanul Huq.

Transformer-Based Text Clustering for Newspaper Articles, pages 443–457. 06 2023.