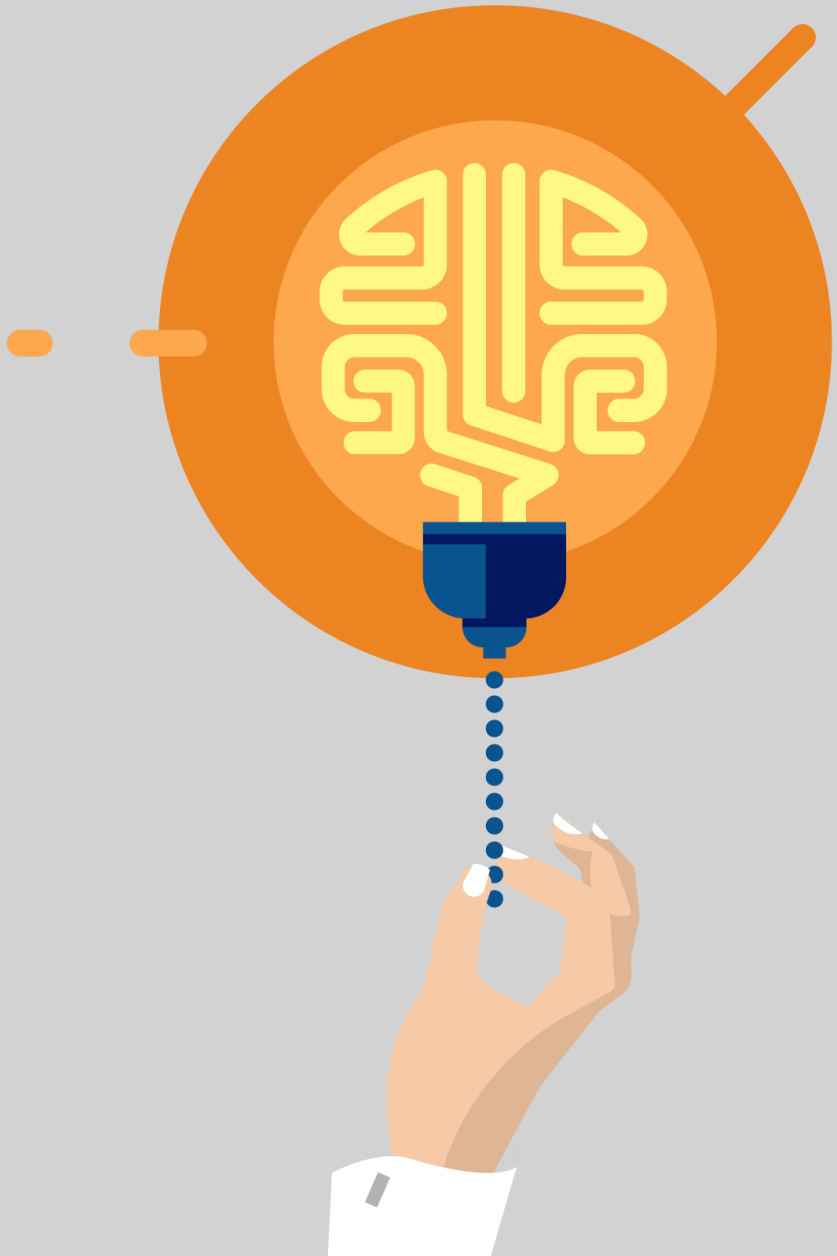
Microsoft

Understanding Code using Natural Language & Graph Neural Networks

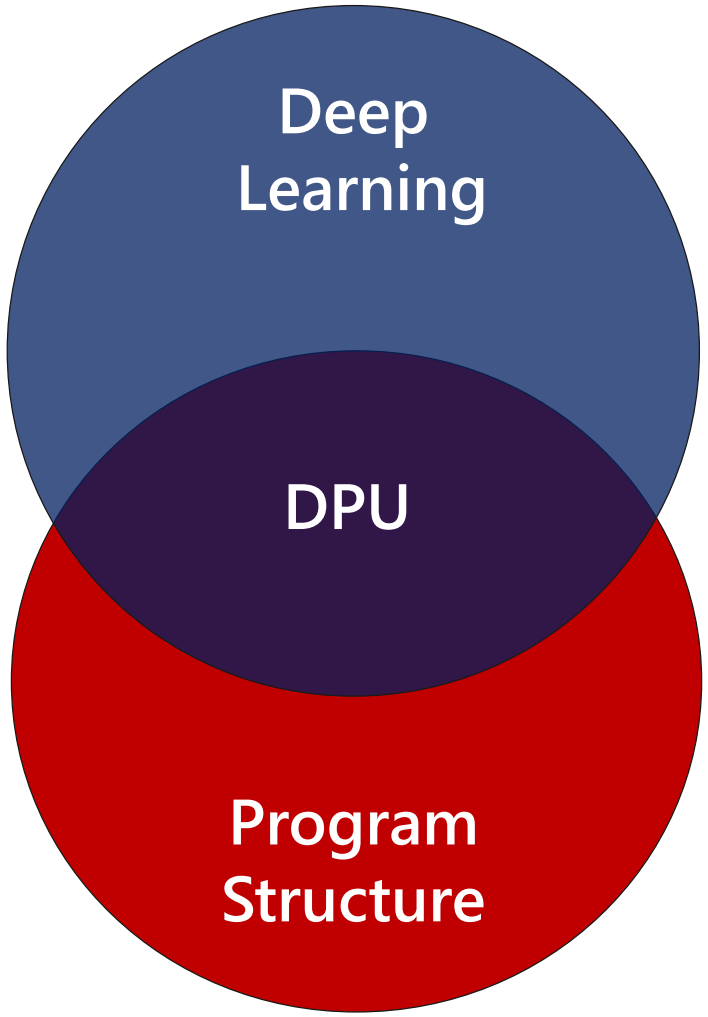
Miltos Allamanis

Microsoft Research Cambridge

Joint work with Marc Brockschmidt, Alex Gaunt, Alex Polozov, Patrick Fernandes, Mahmoud Khademi



Deep Program Understanding



✓ Understands images/language/speech
✓ Finds patterns in noisy data

- Requires many samples
- Handling structured data is hard

✓ Interpretable
✓ Generalisation verifiable

- Manual effort
- Limited to specialists

Microsoft

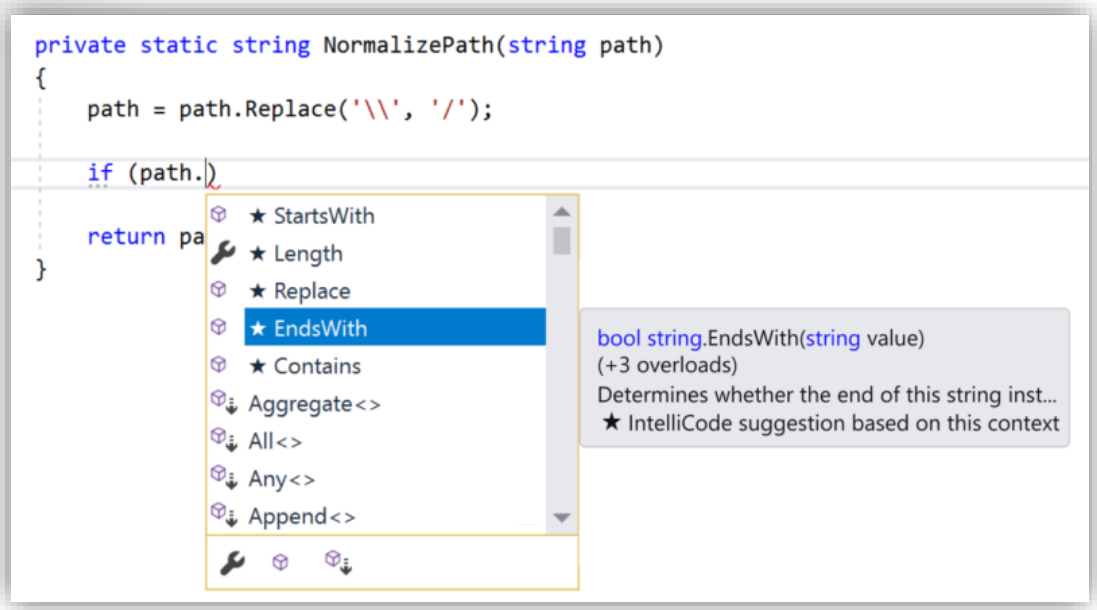
Source Code and Natural Language

Code Autocompletion

```
Text text = new Text(parent, SWT.NONE);
text.
```

setLayoutData(Object layoutData) : void - Control - 86%
setText(String string) : void - Text - 51%
addModifyListener(ModifyListener listener) : void - Text - 31%
getText() : String - Text - 12%
setEnabled(boolean enabled) : void - Control - 11%
setFont(Font font) : void - Control - 8%
Press '^Space' to show Adaptive Template Proposals (Code Recommenders)

http://www.eclipse.org/recommenders/



https://visualstudio.microsoft.com/services/intellicode/

Argument Swapping

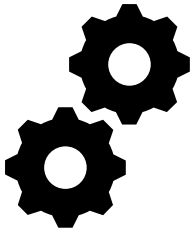
Type	Parameter	Original argument	Correct argument
Duration	responseTTLDuration	frequencyCapDuration	responseTTLDuration
Duration	frequencyCapDuration	responseTTLDuration	frequencyCapDuration
List<A>	slotResponse	slotResponse	slotResponse

Builder	builder	builder	builder
boolean	isTransposed	isTransposed	isTransposed
int	startColumnIndex	a.getStartColumnIndex()	0
int	endColumnIndex	a.getEndColumnIndex()	rows.size()
int	startRow	0	a.getStartColumnIndex()
int	endRow	rows.size()	a.getEndColumnIndex()

Rice *et al.* 2017 “Detecting Argument Selection Defects”

Inferring Type Refinements

Conceptual Types



Defined Types

"a password" → `string password;`

"a JSON string" → `string data = Json.Load();`

Latent; we don't observe in the *conceptual* types.

Defined explicitly by the programmer.

Dash *et al.* 2018 "RefiNym: Using Names to Refine Types"

Variable Misuse

```
// Create or update the document.
var newDocument = await cosmosClient.UpsertDocumentAsync(cosmosDbCollectionUri, document);

if (updateRecord)
{
    logger.WriteLog($"Updated {existingDocument} to {newDocument}");
}
else
{
    logger.WriteLog($"Added {existingDocument}");
```

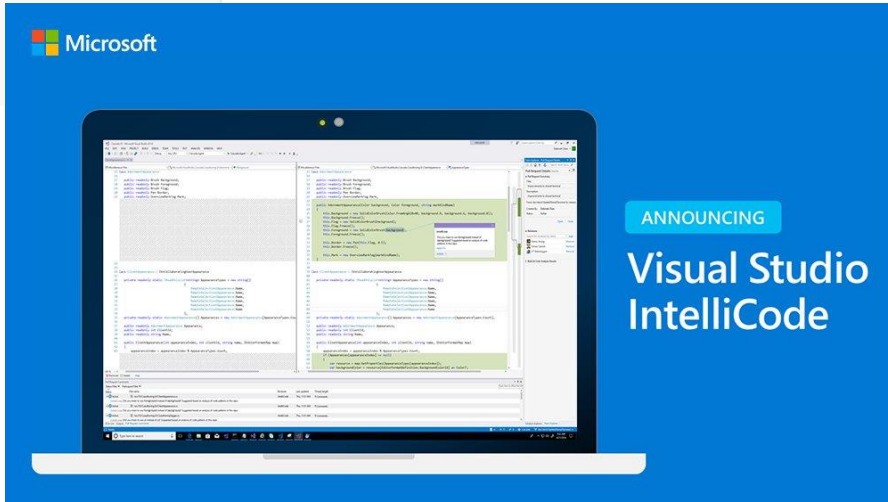
[redacted] Update 1 1 Resolved

Based on this repo's code patterns, did you intend to use 'newDocument' (confidence 92%) rather than 'existingDocument` (confidence 7%) here? Review is recommended by Research bot's Variable Misuse analysis.

JK [redacted]

+1

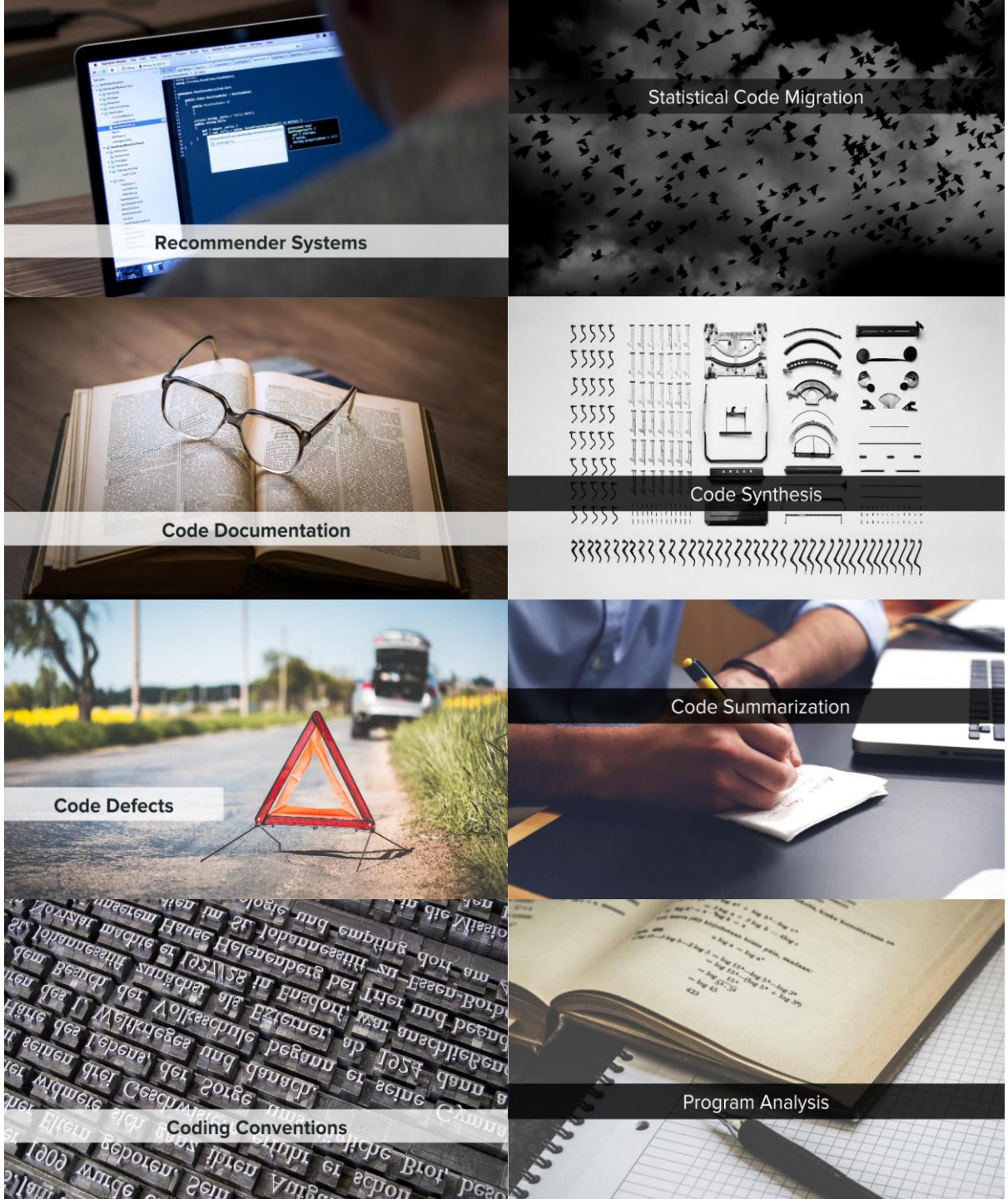
Allamanis *et al.* "Learning to Represent Programs with Graphs". 2018



Research in ML+Code

- Infer latent intent
- Ambiguous information

https://ml4code.github.io



1

A Survey of Machine Learning for Big Code and Naturalness

MILTIADIS ALLAMANIS, Microsoft Research
EARL T. BARR, University College London
PREMKUMAR DEVANBU, University of California, Davis
CHARLES SUTTON, University of Edinburgh and The Alan Turing Institute

Research at the intersection of machine learning, programming languages, and software engineering has recently taken important steps in proposing learnable probabilistic models of source code that exploit code's abundance of patterns. In this article, we survey this work. We contrast programming languages against natural languages and discuss how these similarities and differences drive the design of probabilistic models. We present a taxonomy based on the underlying design principles of each model and use it to navigate the literature. Then, we review how researchers have adapted these models to application areas and discuss cross-cutting and application-specific challenges and opportunities.

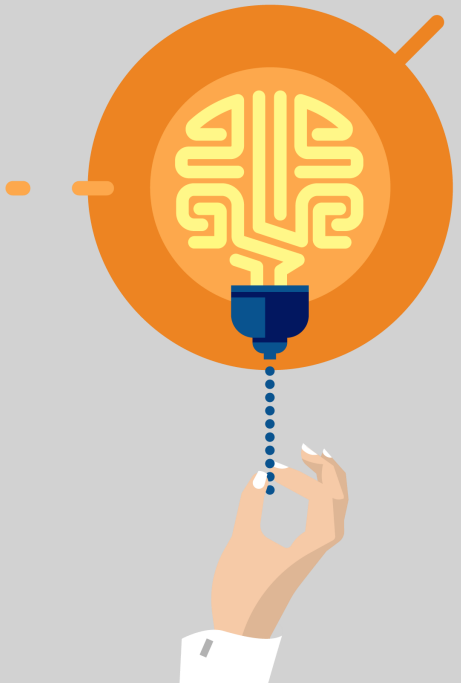
CCS Concepts: • Computing methodologies → Machine learning; Natural language processing; • Soft-

Microsoft

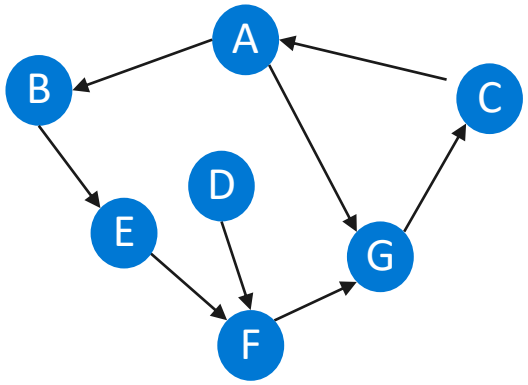
- Graph Neural Networks
- Applications to Source Code
- Applications to NLP

Graph Neural Networks

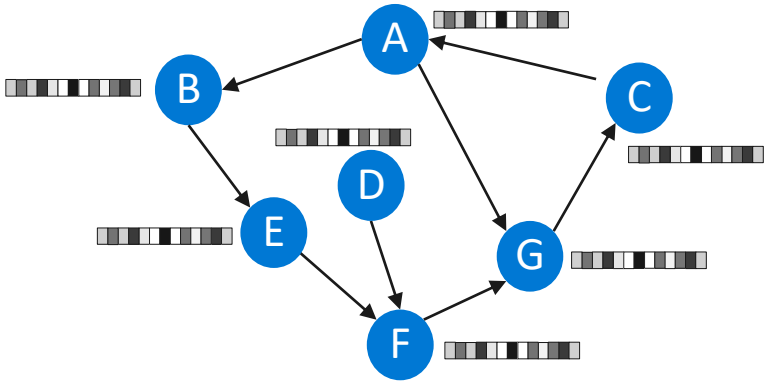
and Neural Message Passing



Graph Neural Networks



Graph Representation
of Problem

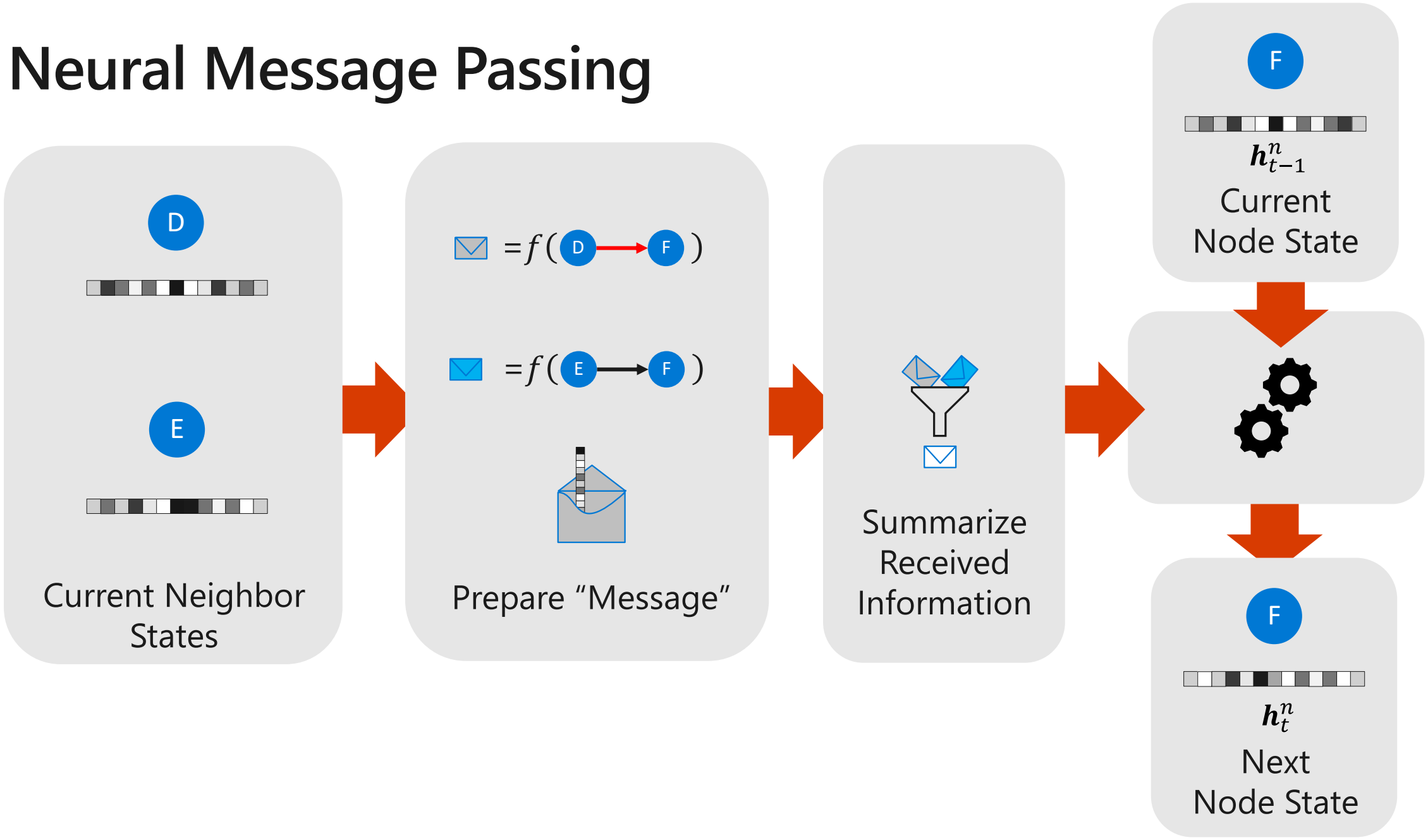


Initial Representation
of each node

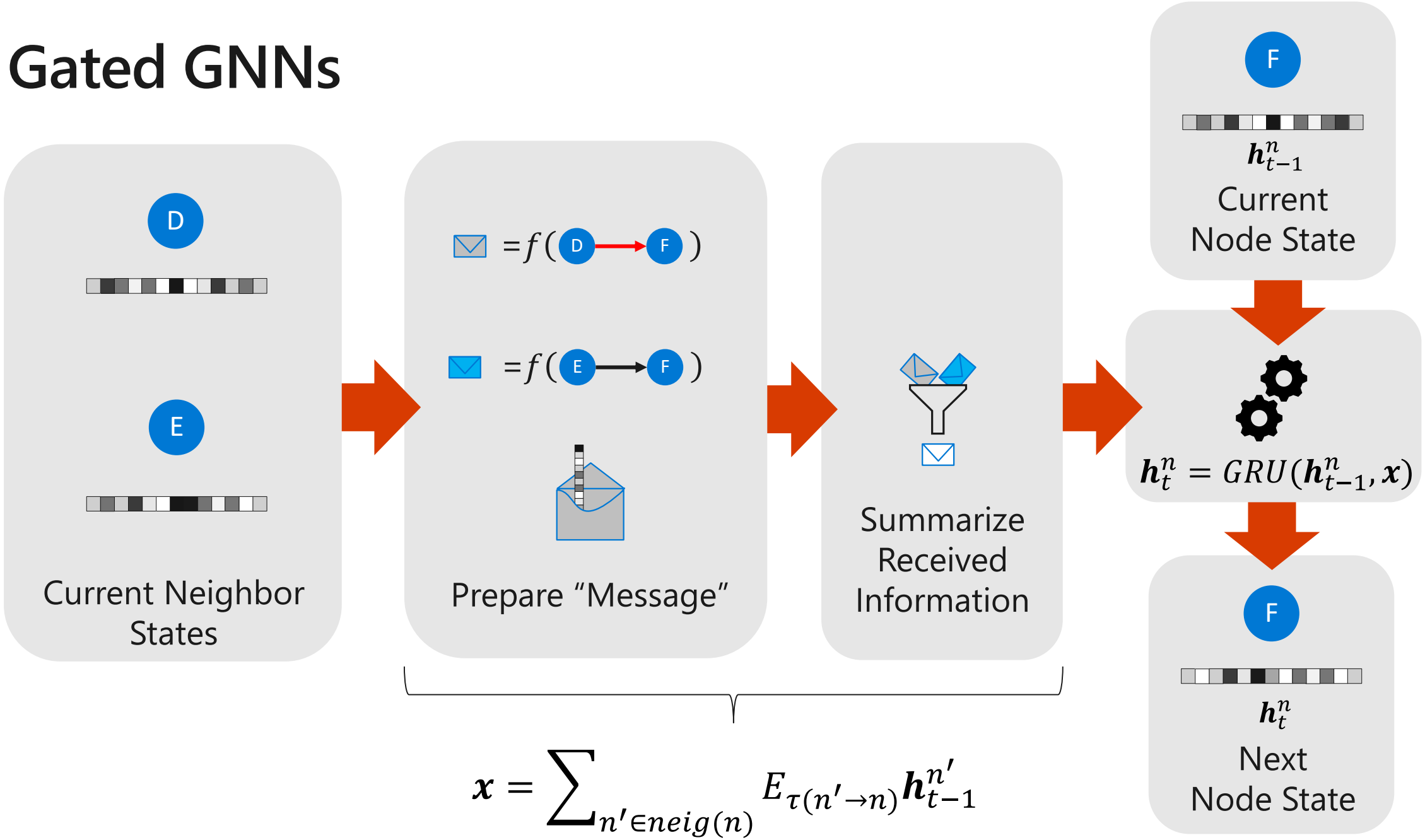
Li et al (2015). Gated Graph Sequence Neural Networks.

Gilmer et al (2017). Neural Message Passing for Quantum Chemistry.

Neural Message Passing

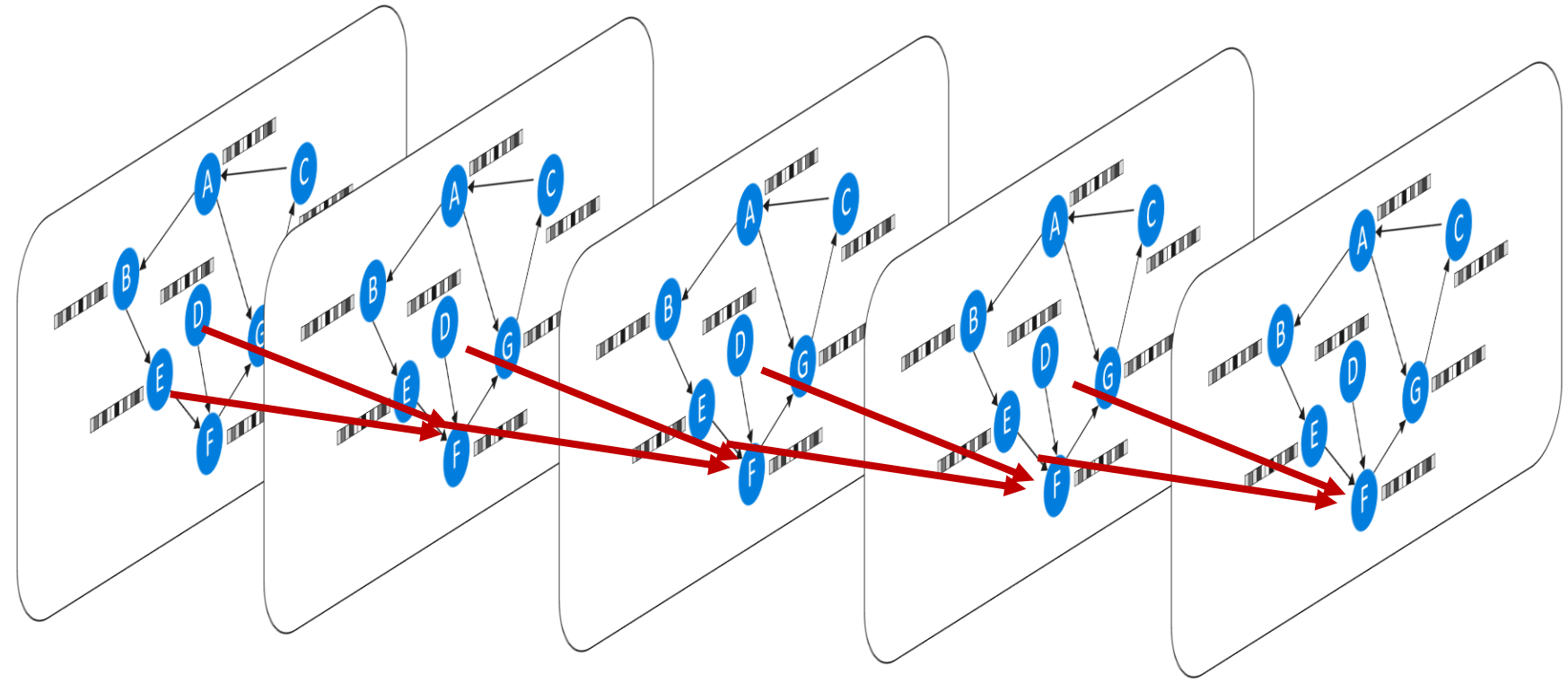


Gated GNNs

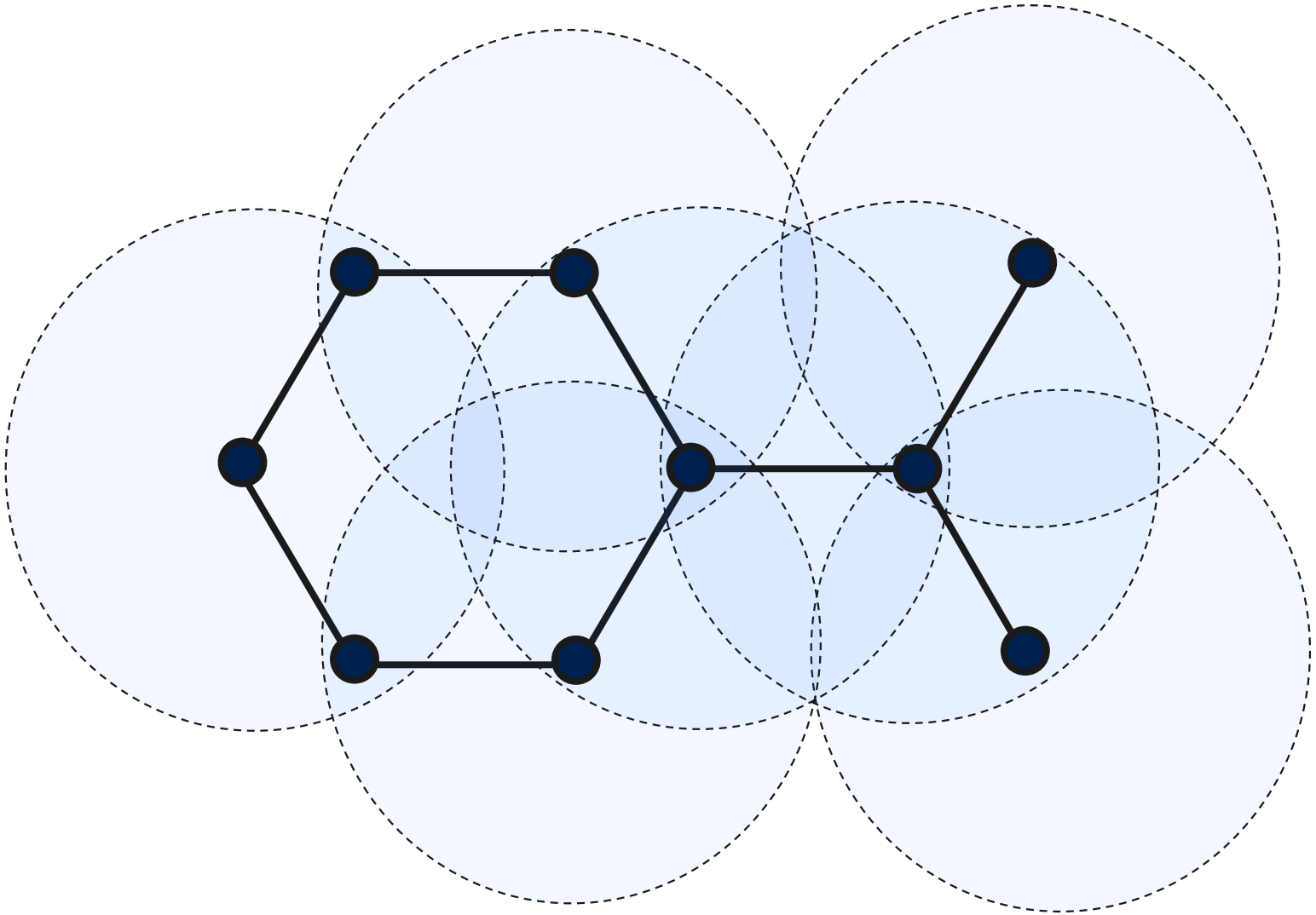


Li et al (2015). Gated graph sequence neural networks.

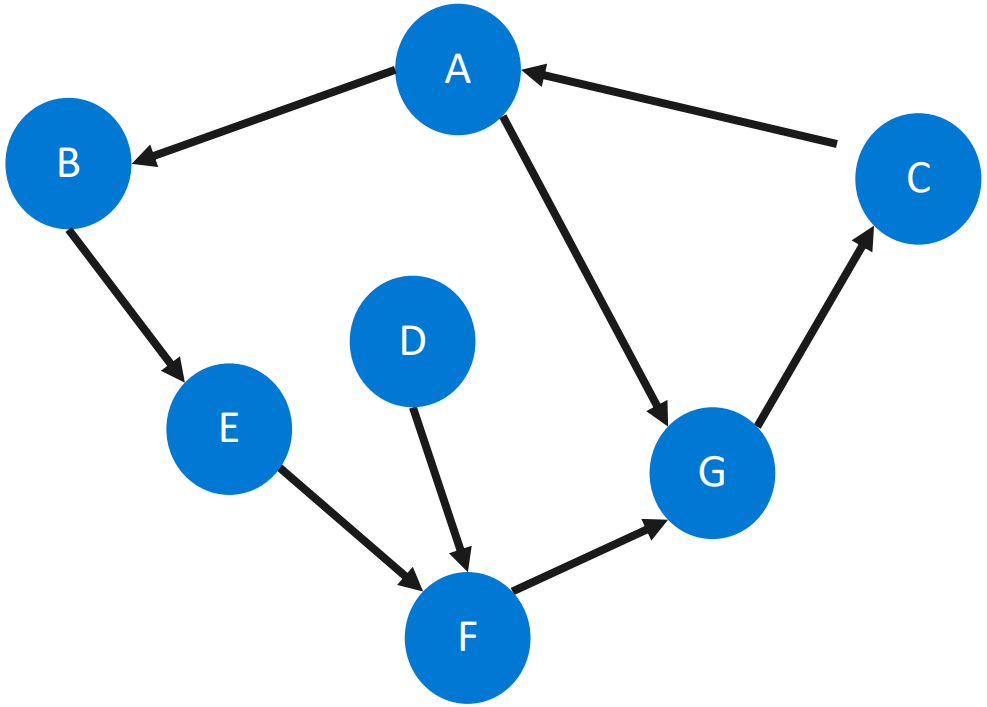
Graph Neural Networks: Message Passing



GNNs: Synchronous Message Passing (All-to-All)



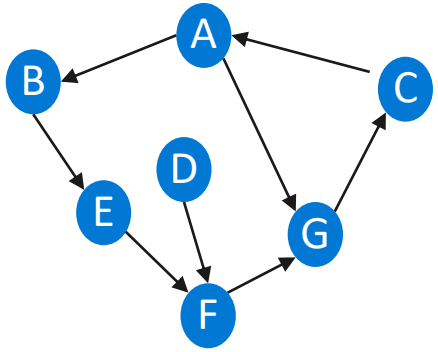
GGNs: Asynchronous Message Passing

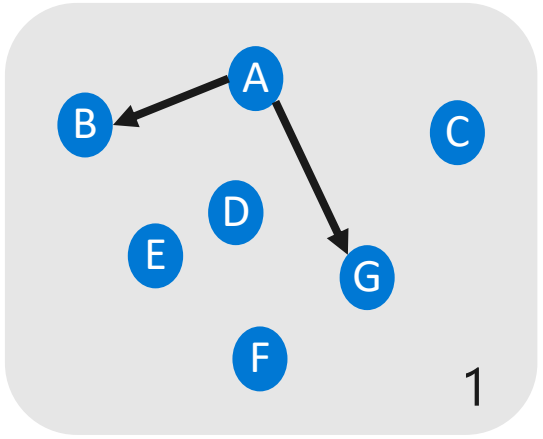


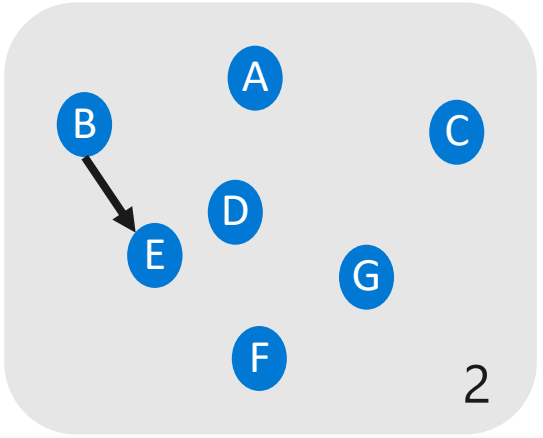
Define Schedule

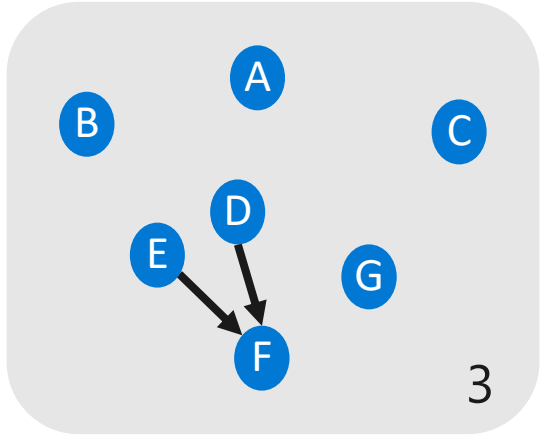
Send Messages

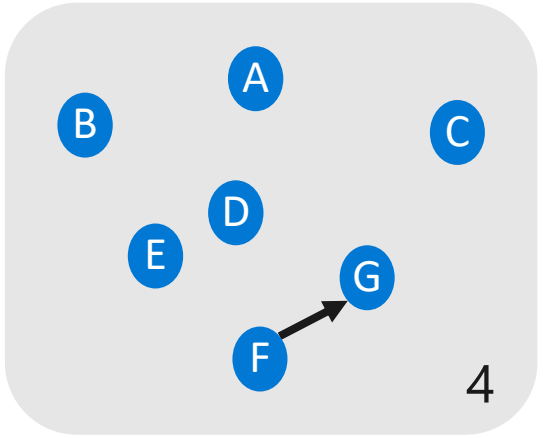
GGNs: Asynchronous Message Passing

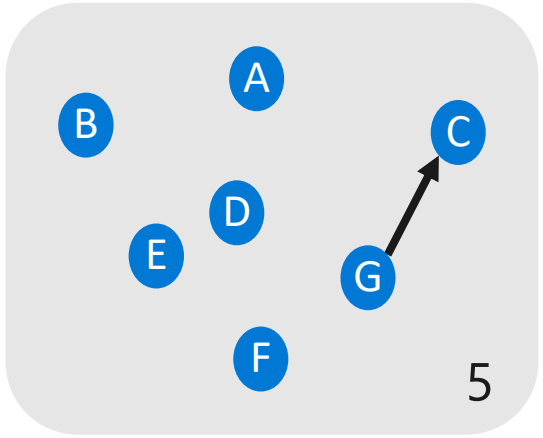


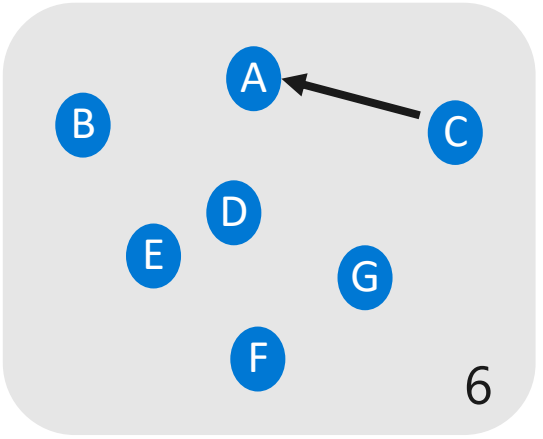




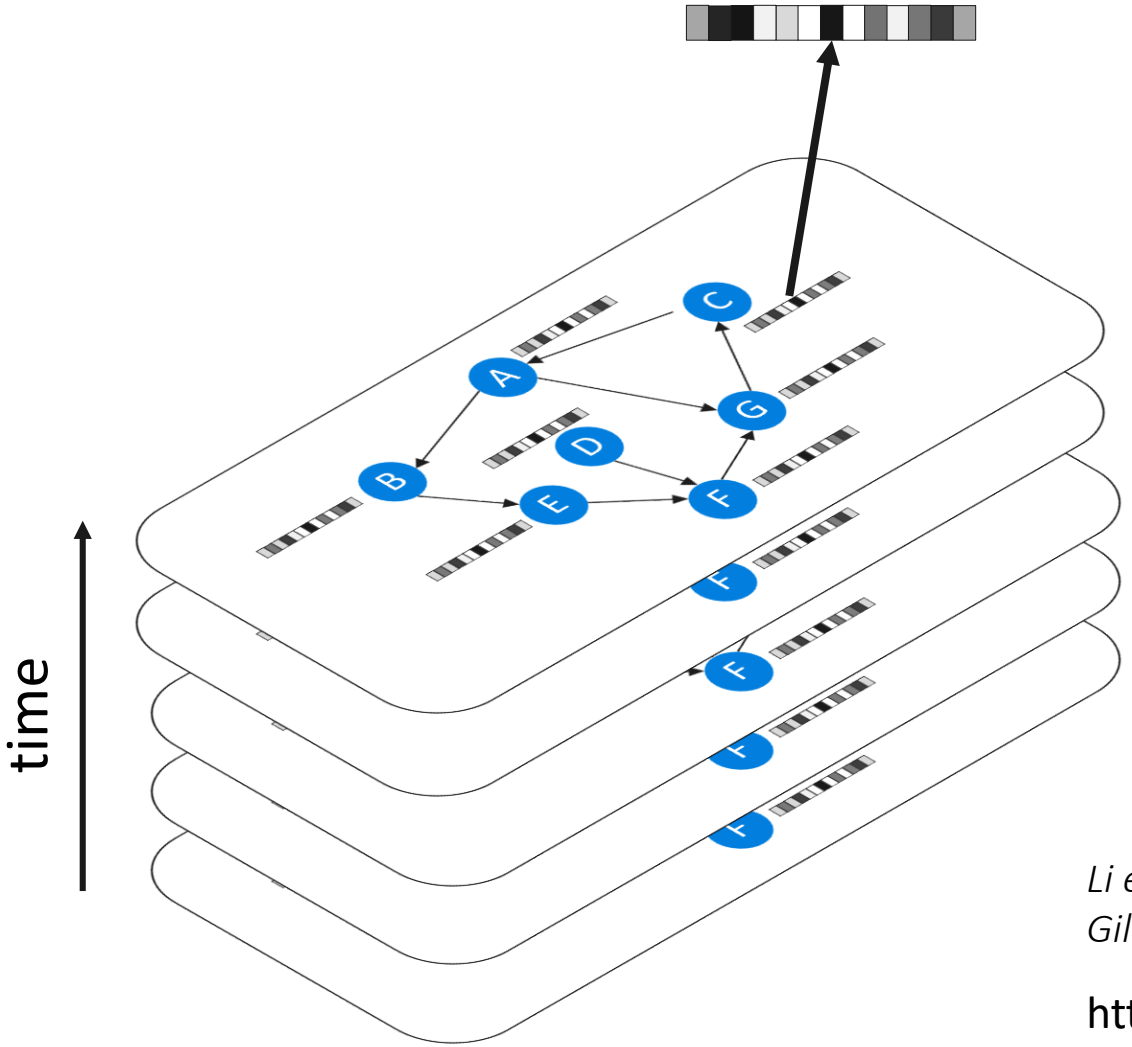








Graph Neural Networks: Output



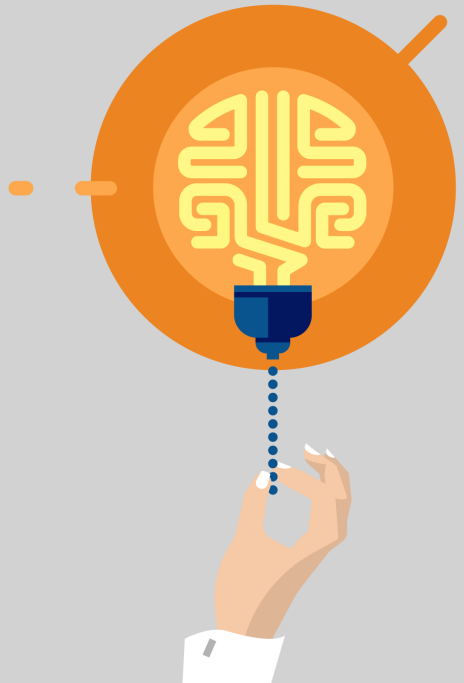
- node selection
- node classification
- graph classification

Li et al (2015). Gated Graph Sequence Neural Networks.
Gilmer et al (2017). Neural Message Passing for Quantum Chemistry.

https://github.com/Microsoft/gated-graph-neural-network-samples

Understanding & Generating Source Code

…with graph neural networks.

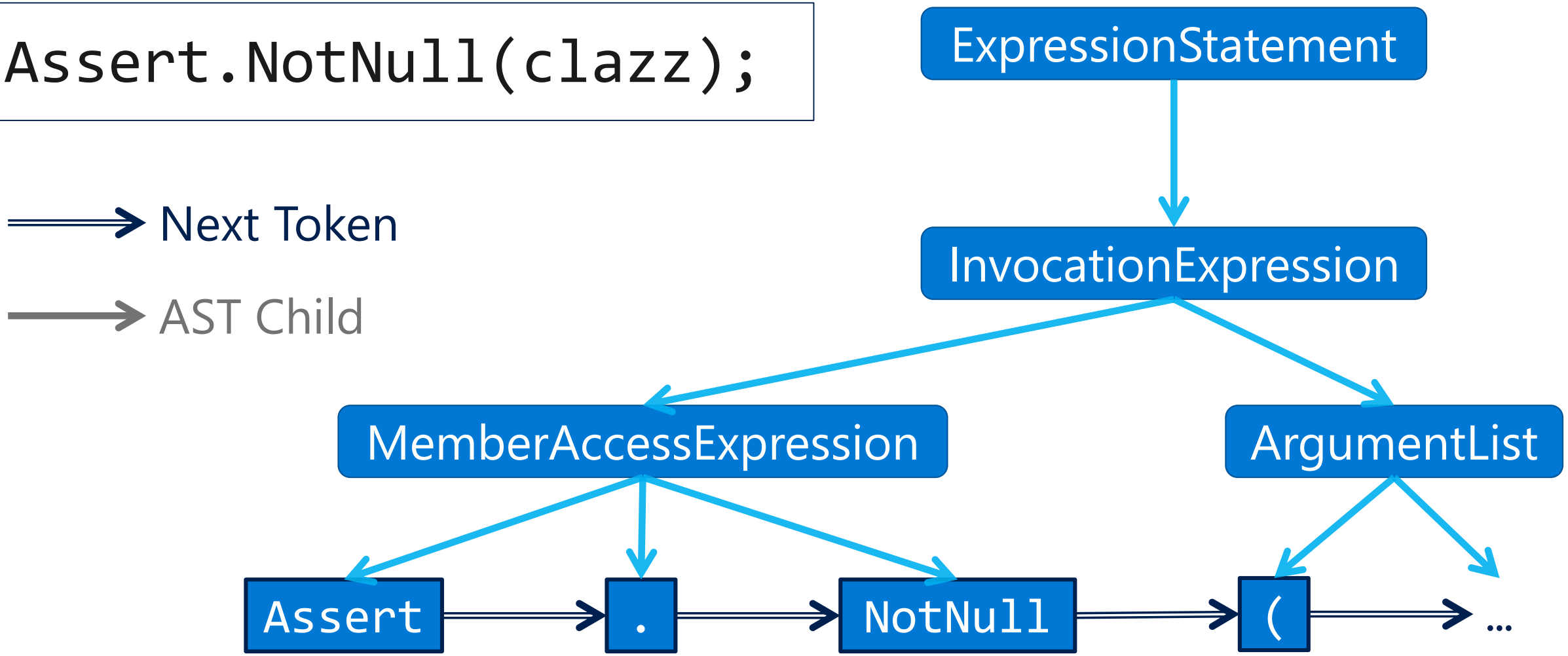


Programs as Graphs

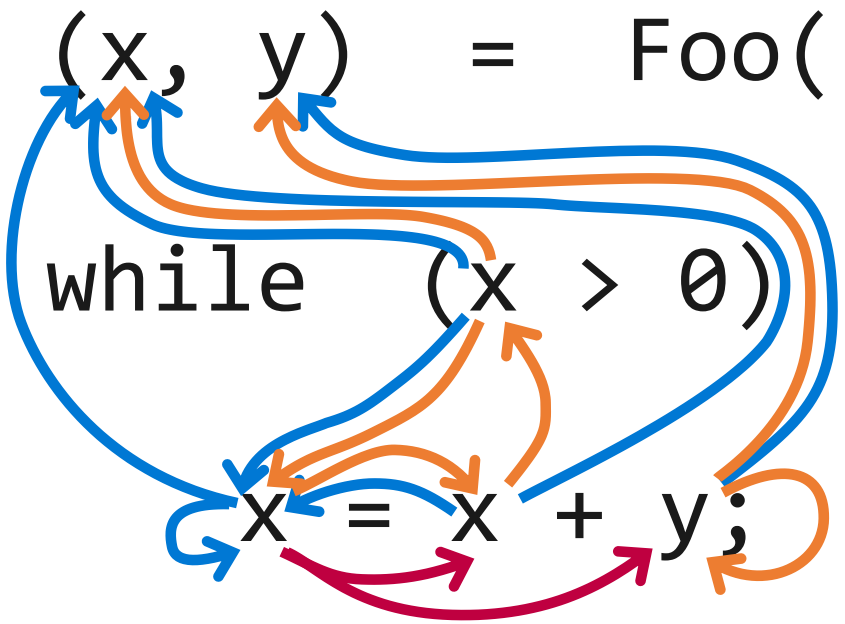
```
int SumPositive(int[] arr, int lim) {
  int sum = 0;
  for (int i = 0; i < lim; i++)
    if (arr[i] > 0)
        sum += arr[i];

  return sum;
}
```

Programs as Graphs: Syntax

```
Assert.NotNull(clazz);
```



Programs as Graphs: Data Flow



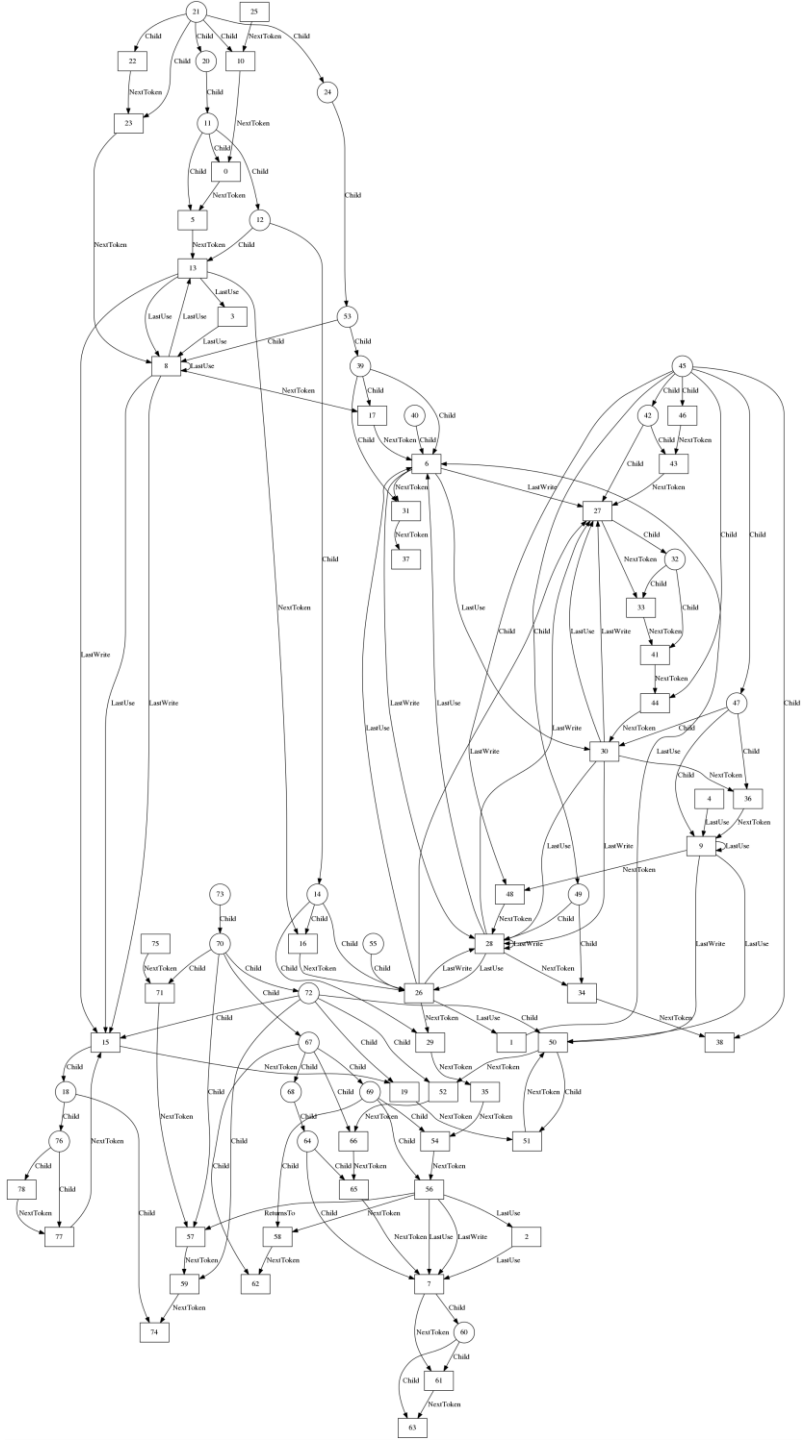
Last Write

Last Use

Computed From

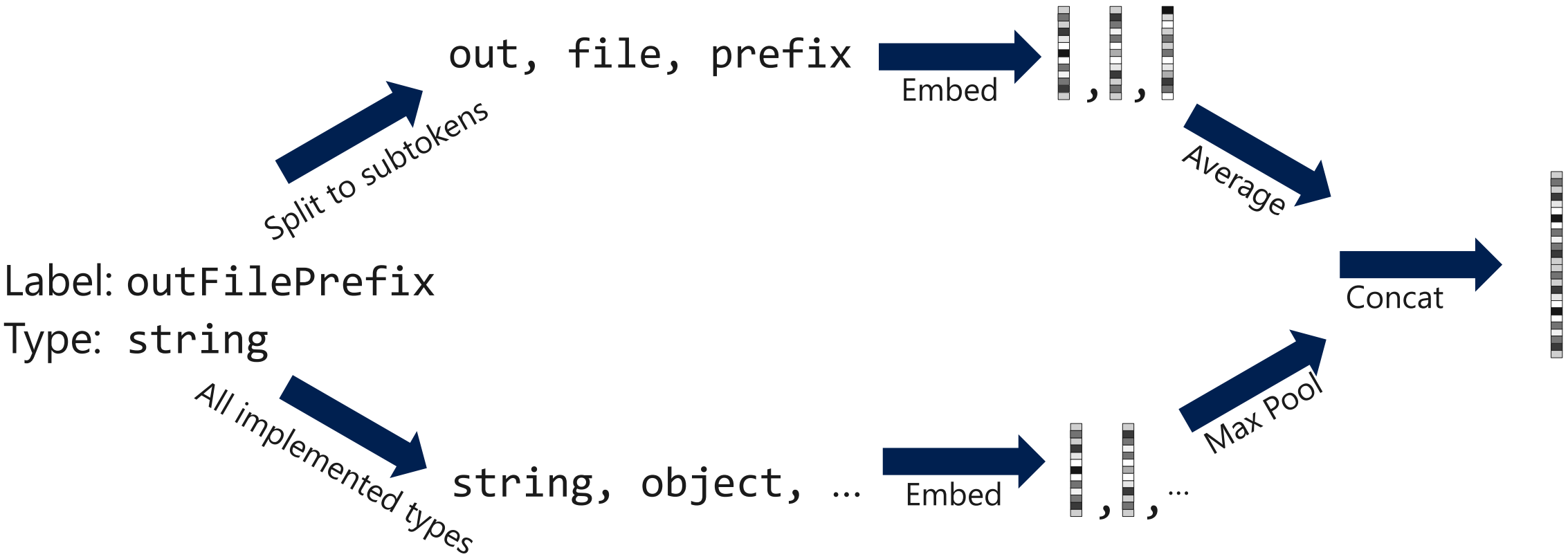
Programs as Graphs

```
int SumPositive(int[] arr, int lim) {
  int sum = 0;
  for (int i=0; i<; ++)
    if ([]>0)
      +=[];
  return sum;
}
```



~900 nodes/graph ~8k edges/graph

Initial Node Representations



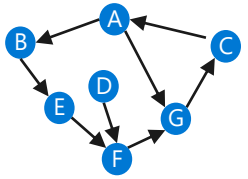
Variable Misuse Task

```
var clazz=classTypes["Root"].Single() as JsonCodeGenerator.ClassType;
Assert.NotNull(clazz);

var first=classTypes["RecClass"].Single() as JsonCodeGenerator.ClassType;
Assert.NotNull(      );

Assert.Equal("string", first.Properties["Name"].Name);
Assert.False(clazz.Properties["Name"].IsArray);
```

Possible type-correct options: `clazz, first`

Not easy to catch with static analysis tools.

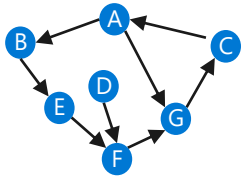
Graph Representation for Variable Misuse




```
var clazz=classTypes["Root"].Single() as JsonCodeGenerator.ClassType;
Assert.NotNull(clazz);

var first=classTypes["RecClass"].Single() as JsonCodeGenerator.ClassType;
Assert.NotNull(      );

Assert.Equal("string", first.Properties["Name"].Name);
Assert.False(clazz.Properties["Name"].IsArray);
```

Possible type-correct options: `clazz, first`

Graph Representation for Variable Misuse




```
var clazz=classTypes["Root"].Single() as JsonCodeGenerator.ClassType;
Assert.NotNull(clazz);

var first=classTypes["RecClass"].Single() as JsonCodeGenerator.ClassType;
Assert.NotNull( SLOT );

Assert.Equal("string", first.Properties["Name"].Name);
Assert.False(clazz.Properties["Name"].IsArray);
```

first

clazz

Goal: make the representation of SLOT as close as possible to the representation of the correct candidate node

$$f\left(\boldsymbol{h}_T^{SLOT}, \boldsymbol{h}_T^{first}\right) \gg f\left(\boldsymbol{h}_T^{SLOT}, \boldsymbol{h}_T^{clazz}\right)$$

Dataset

2.9MLOC

GitHub

Name	Git SHA	kLOCs	Slots	Vars	Description
Akka.NET	719335a1	240	51.3k	51.2k	Actor-based Concurrent & Distributed Framework
AutoMapper	2ca7c2b5	46	3.7k	10.7k	Object-to-Object Mapping Library
BenchmarkDotNet	1670ca34	28	5.1k	6.1k	Benchmarking Library
BotBuilder	190117c3	44	6.4k	8.7k	SDK for Building Bots
choco	93985688	36	3.8k	5.2k	Windows Package Manager
commandline†	09677b16	11	1.1k	2.3k	Command Line Parser
CommonMark.NET^{Dev}	f3d54530	14	2.6k	1.4k	Markdown Parser
Dapper	931c700d	18	3.3k	4.7k	Object Mapper Library
EntityFramework	fa0b7ec8	263	33.4k	39.3k	Object-Relational Mapper
Hangfire	ffc4912f	33	3.6k	6.1k	Background Job Processing Library
Humanizer†	cc11a77e	27	2.4k	4.4k	String Manipulation and Formatting
Lean†	f574bfd7	190	26.4k	28.3k	Algorithmic Trading Engine
Nancy	72e1f614	70	7.5k	15.7	HTTP Service Framework
Newtonsoft.Json	6057d9b8	123	14.9k	16.1k	JSON Library
Ninject	7006297f	13	0.7k	2.1k	Code Injection Library
NLog	643e326a	75	8.3k	11.0k	Logging Library
Opserver	51b032e7	24	3.7k	4.5k	Monitoring System
OptiKey	7d35c718	34	6.1k	3.9k	Assistive On-Screen Keyboard
orleans	e0d6a150	300	30.7k	35.6k	Distributed Virtual Actor Model
Polly	0afdbc32	32	3.8k	9.1k	Resilience & Transient Fault Handling Library
quartznet	b33e6f86	49	9.6k	9.8k	Scheduler
ravendb^{Dev}	55230922	647	78.0k	82.7k	Document Database
RestSharp	70de357b	20	4.0k	4.5k	REST and HTTP API Client Library
Rx.NET	2d146fe5	180	14.0k	21.9k	Reactive Language Extensions
scriptcs	f3cc8bcb	18	2.7k	4.3k	C# Text Editor
ServiceStack	6d59da75	231	38.0k	46.2k	Web Framework
ShareX	718dd711	125	22.3k	18.1k	Sharing Application
SignalR	fa88089e	53	6.5k	10.5k	Push Notification Framework
Wox	cdaf6272	13	2.0k	2.1k	Application Launcher

Quantitative Results – Variable Misuse

Accuracy (%)	BiGRU	BiGRU + Dataflow	GGNN
Seen Projects	50.0	73.7	**85.5**

Seen Projects: 24 F/OSS C# projects (2060 kLOC): Used for train and test

3.8 type-correct alternative variables per slot (median 3, σ= 2.6)

Quantitative Results – Variable Misuse

Accuracy (%)	BiGRU	BiGRU + Dataflow	GGNN
Seen Projects	50.0	73.7	**85.5**
Unseen Projects	28.9	60.2	**78.2**

Seen Projects: 24 F/OSS C# projects (2060 kLOC): Used for train and test
Unseen Projects: 3 F/OSS C# projects (228 kLOC): Used only for test
3.8 type-correct alternative variables per slot (median 3, $\sigma = 2.6$)

```
        // Create or update the document.
        var newDocument = await cosmosClient.UpsertDocumentAsync(cosmosDbCollectionUri, document);

        if (updateRecord)
        {
            logger.WriteLog($"Updated {existingDocument} to {newDocument}");
        }
        else
        {
            logger.WriteLog($"Added {existingDocument}");
```

[redacted] Update 1

1 Resolved

Based on this repo's code patterns, did you intend to use 'newDocument' (confidence 92%) rather than 'existingDocument` (confidence 7%) here? Review is recommended by Research bot's Variable Misuse analysis.

JK [redacted]

+1

Code Summarization to Natural Language

Code Function

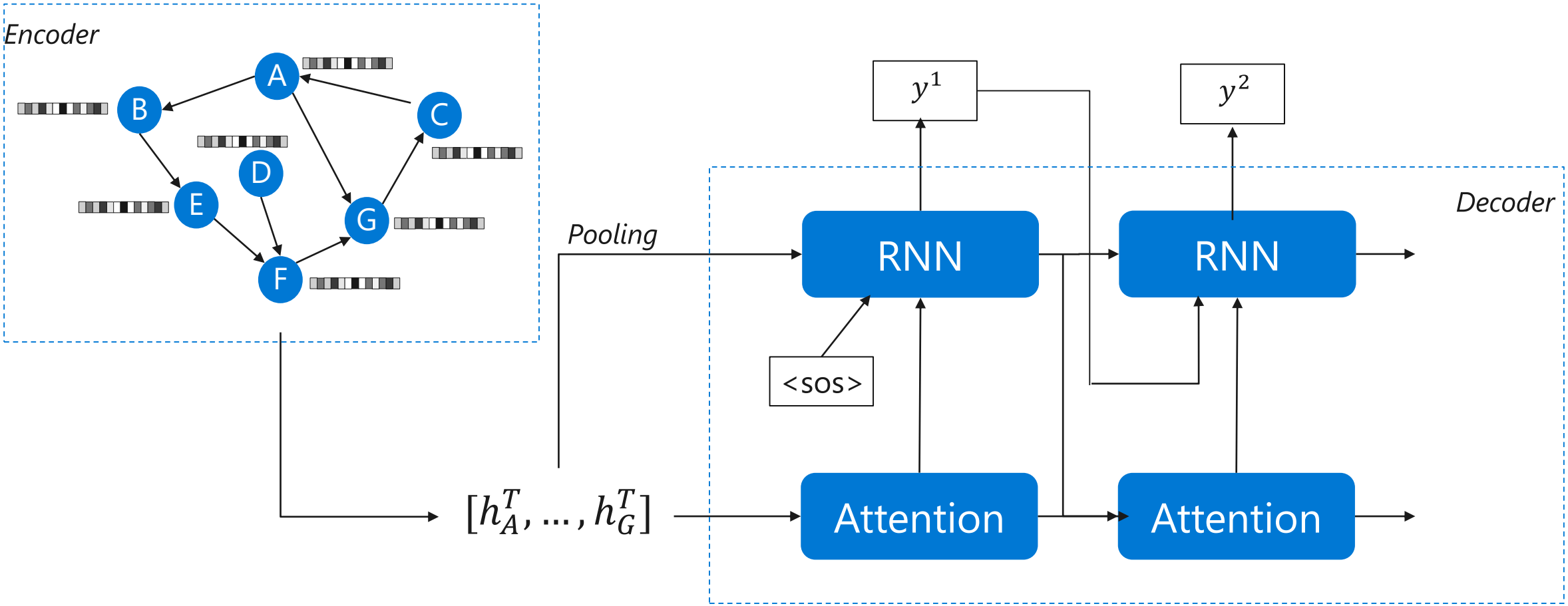
```
int SumPositive(int[] arr, int lim) {
  int sum = 0;
  for (int i = 0; i < lim; i++)
    if (arr[i]>0)
        sum += arr[i];

  return sum;
}
```

Summary

Returns the sum of the positive numbers in an array

Fernandes *et al. 2019* "Neural Structured Summarization"

Graph to Sequence Model



Quantitative Results

C# Documentation	F1	ROUGE-2	ROUGE-L
biRNN -> RNN	35.2	20.8	36.7
GNN -> RNN	38.9	25.6	37.1
biRNN + GNN -> RNN	**45.4**	**28.3**	**41.1**

Java Method Naming	F1	ROUGE-2	ROUGE-L
Alon et al. *	43.0	-	-
biRNN -> RNN	35.8	17.9	39.7
biRNN + GNN -> RNN	**44.7**	**21.1**	**43.1**

** Alon et al. (2018). Code2seq: Generating sequences from structured representations of code.*

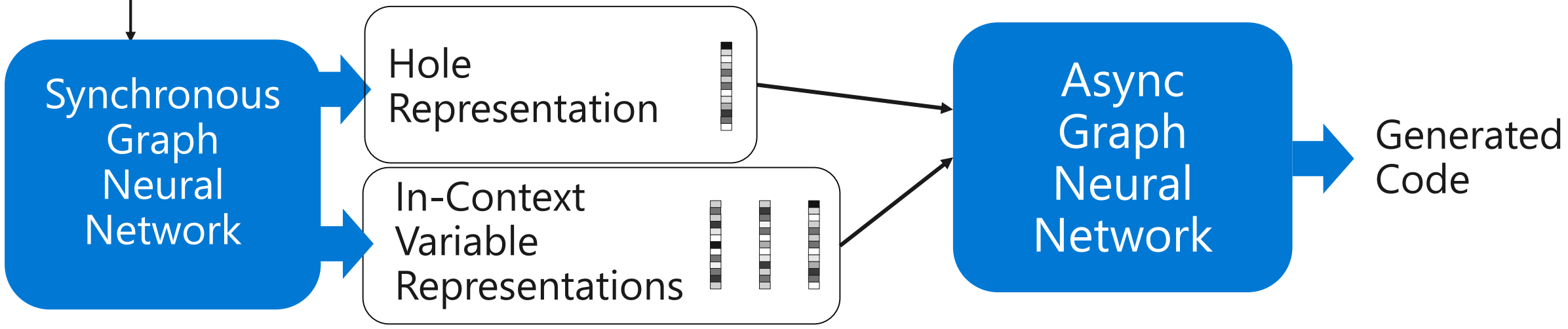
Expression Completion Task

```
int methParamCount = 0;
if (paramCount > 0) {
  IParameterTypeInformation[] moduleParamArr =
    GetParamTypeInformations(Dummy.Signature, paramCount);
  methParamCount = moduleParamArr.Length;
}
if (                                    ) {
  IParameterTypeInformation[] moduleParamArr =
    GetParamTypeInformations(Dummy.Signature,
                             paramCount - methParamCount);
}
```

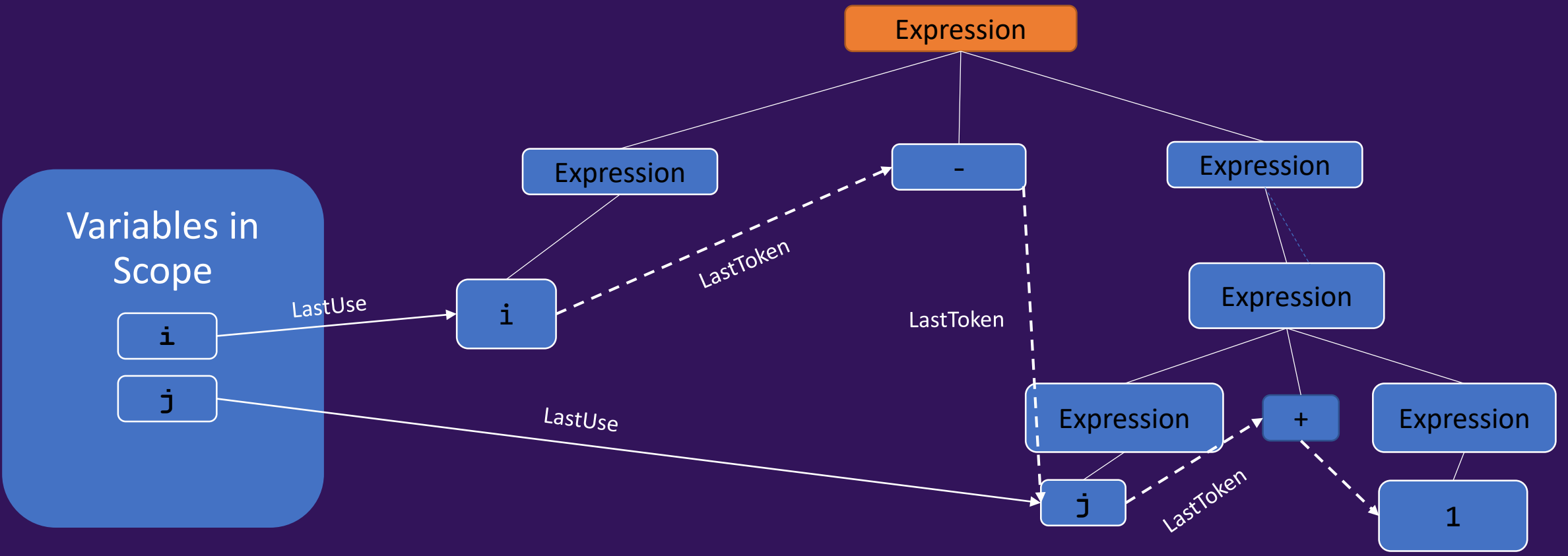
Expression Completion Task

```
int methParamCount = 0;
if (paramCount > 0) {
  IParameterTypeInformation[] moduleParamArr =
    GetParamTypeInformations(Dummy.Signature, paramCount);
  methParamCount = moduleParamArr.Length;
}
if (                                        ) {
  IParameterTypeInformation[] moduleParamArr =
    GetParamTypeInformations(Dummy.Signature,
                             paramCount - methParamCount);
}
```

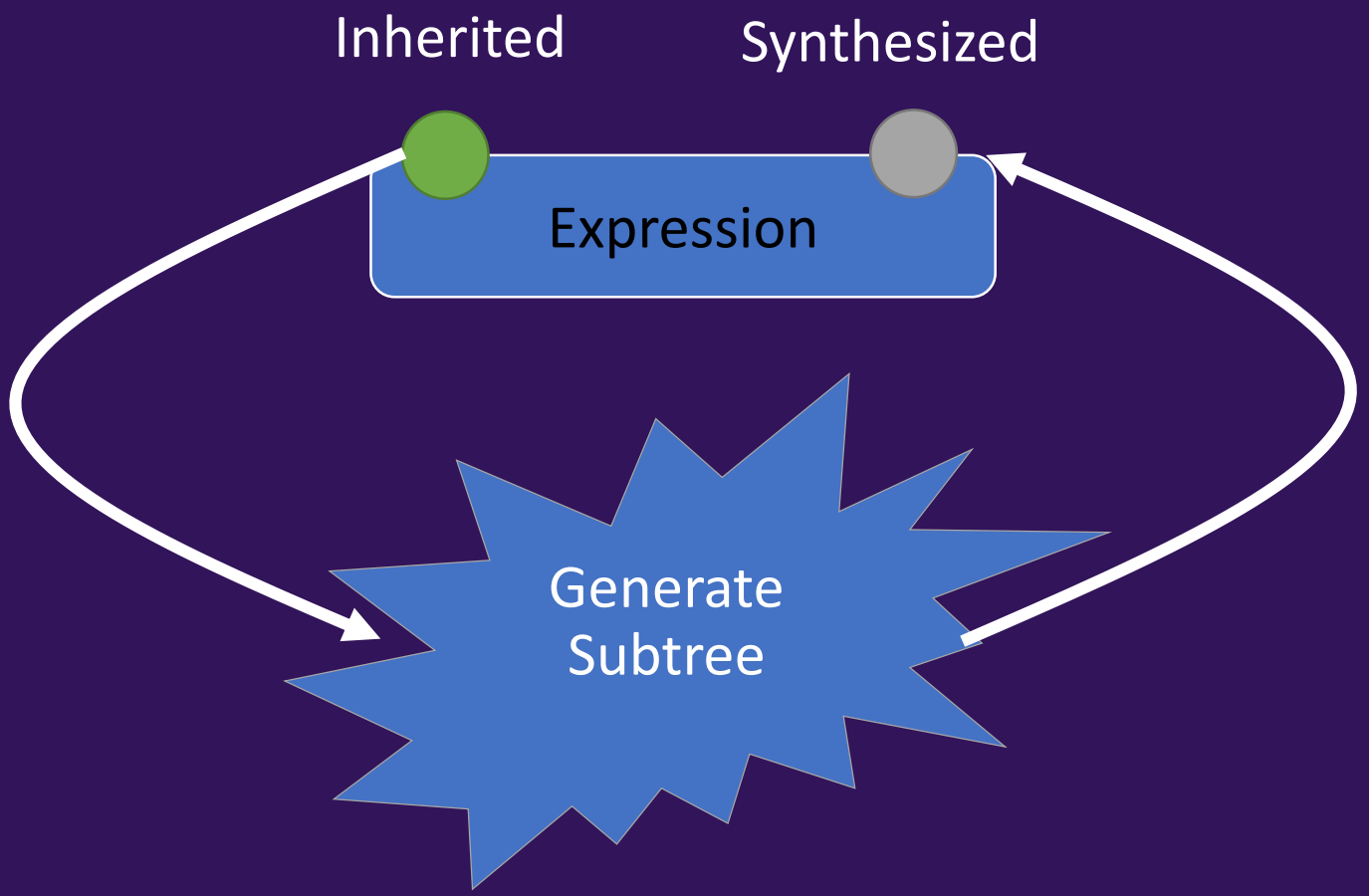
Code Context



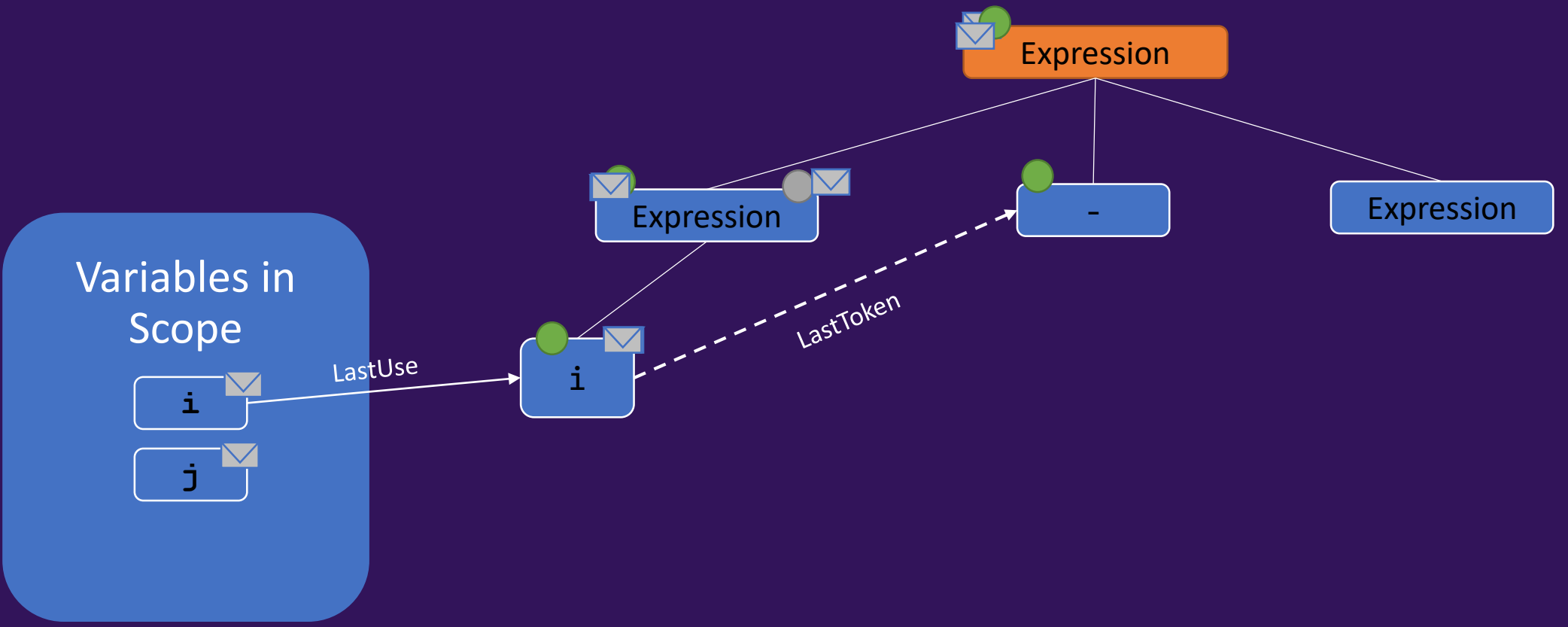
Adding Edges During Generation



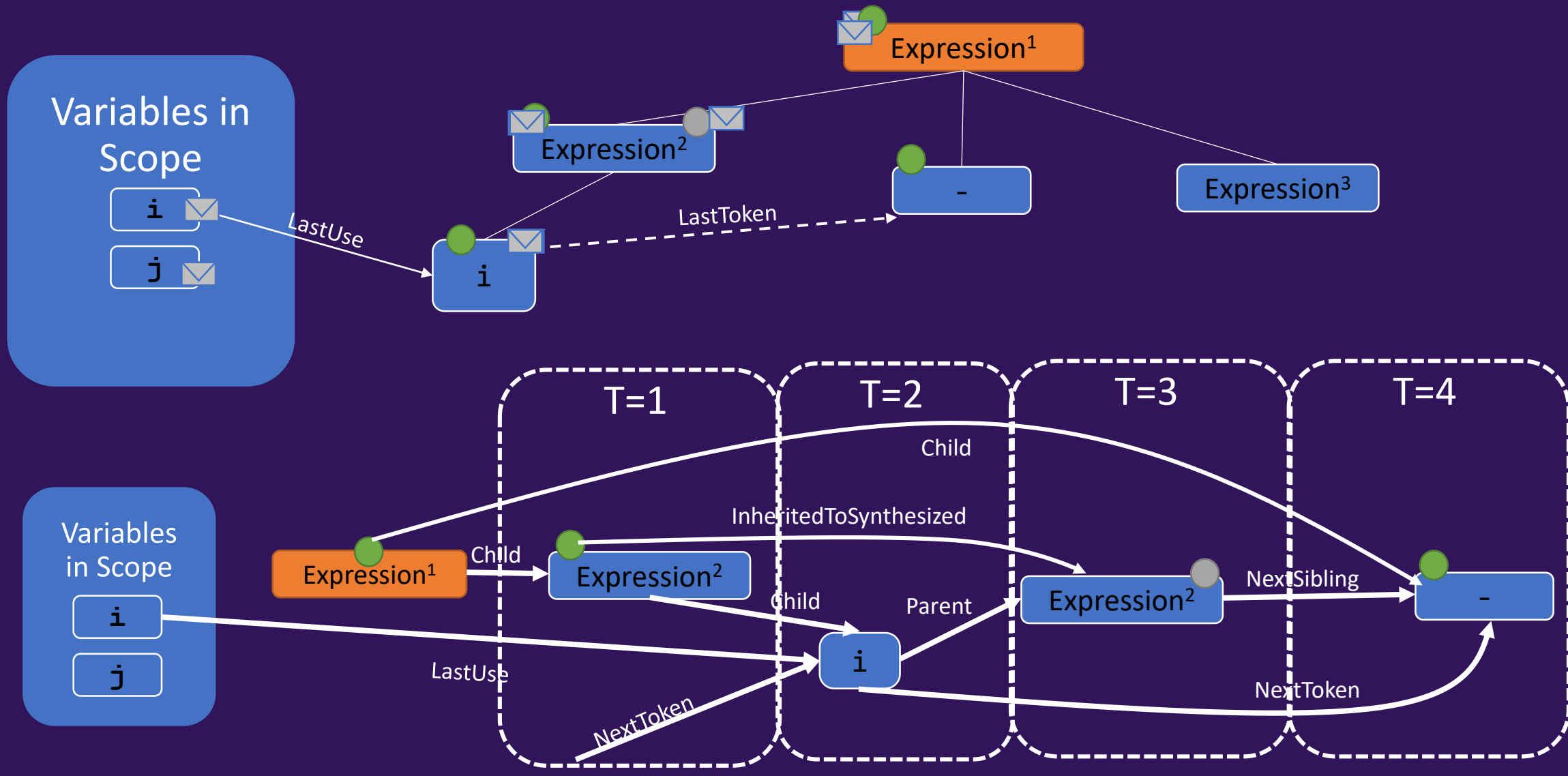
Neural Attribute Grammars (NAG)



Information Propagation on Graph Generation



Information Propagation on Graph Generation



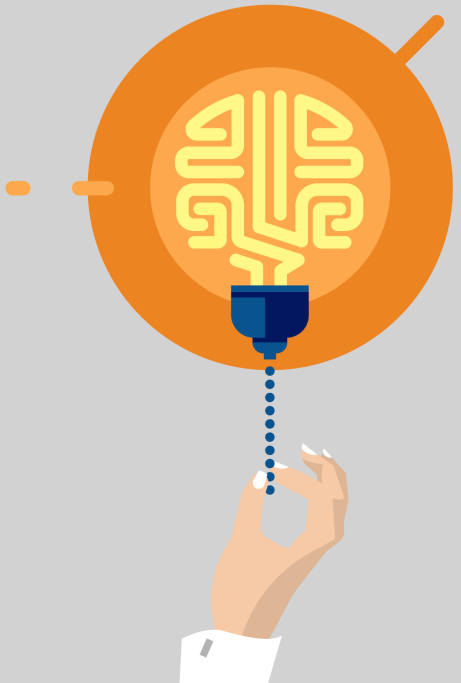
Filling in the Blanks: Quantitative Results

Model	PPL	Type-Correct	Match@1	Match @5
Seq → NAG	8.38	40.4	8.4	15.8
Graph → Tree	5.37	41.2	19.9	36.8
Graph → Syntax Networks	3.03	74.7	32.4	48.1
Graph → Sequentalised Tree	3.48	84.5	36.0	52.7
Graph → Neural Attr. Gram.	3.07	84.5	38.8	57.0

Training data: 479 C# projects from GitHub
Test data: 114 C# projects from GitHub (~100 000 samples)

Natural Language Processing

…with graph neural networks.



Summarization

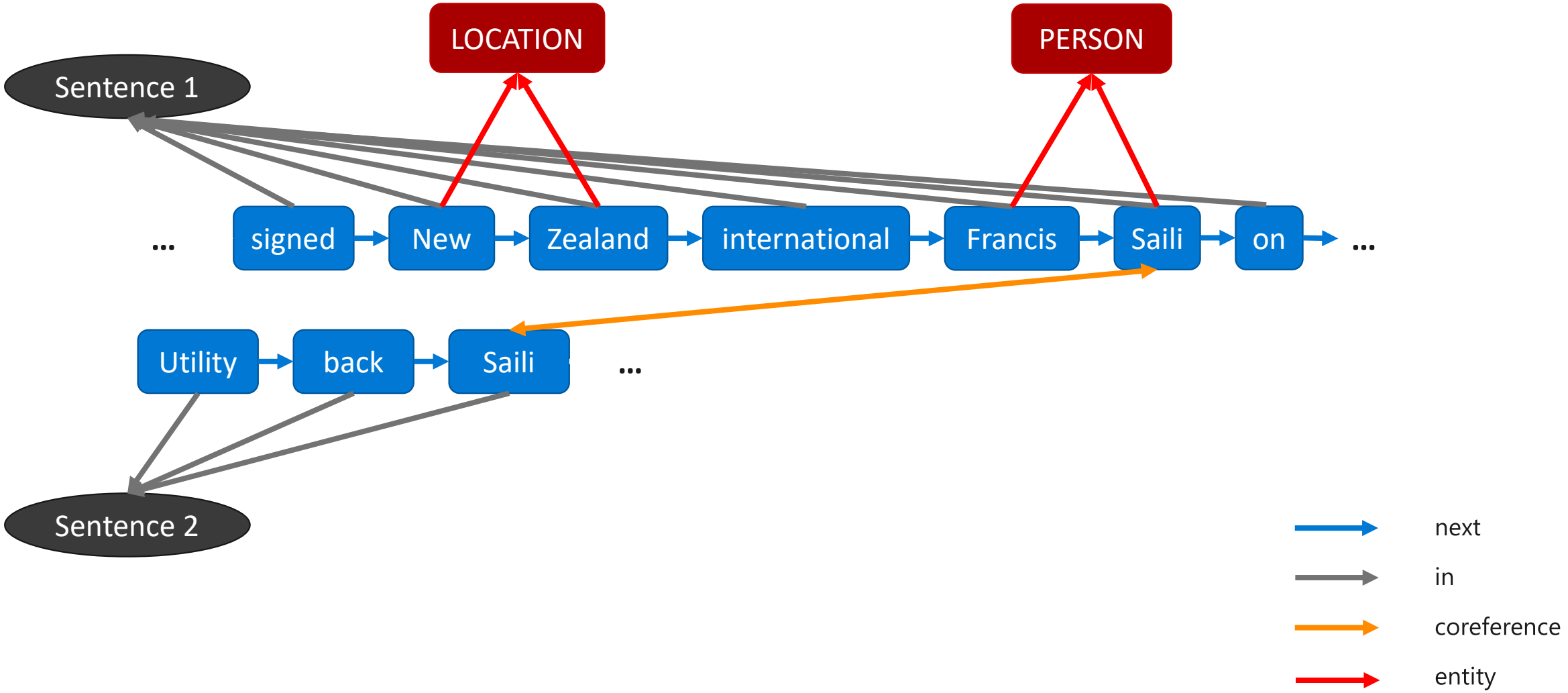
News Article

Gunshots were fired at rapper Lil Wayne ’s tour bus early Sunday in Atlanta. No one was injured in the shooting, and no arrests have been made, Atlanta Police spokeswoman Elizabeth Espy said. Police are still looking for suspects. Officers were called to a parking lot in Atlanta ’s Buckhead neighbourhood, Espy said. They arrived at 3:25 a.m. and located two tour buses that had been shot multiple times. The drivers of the buses said the incident occurred on Interstate 285 near Interstate 75, Espy said. Witnesses provided a limited description of the two vehicles suspected to be involved: a “Corvette-style vehicle” and an SUV. Lil Wayne was in Atlanta for a performance at Compound nightclub Saturday night. CNN’s Carma Hassan contributed to this report.

Summary

Rapper Lil Wayne not injured after shots fired at his tour bus on an Atlanta interstate, police say. No one has been arrested in the shooting.

Structure in Natural Language



Quantitative Results

CNN/DailyMail	ROUGE-1	ROUGE-2	ROUGE-L
RNN -> RNN *	31.3	11.8	**28.8**
RNN + GNN -> RNN	**33.0**	**13.3**	28.3

Encoder -> Decoder

Higher is better

** See et al. (2015). Get To The Point: Summarization with Pointer-Generator Networks*

Quantitative Results

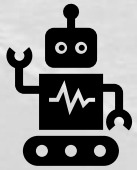
CNN/DailyMail	ROUGE-1	ROUGE-2	ROUGE-L
RNN -> RNN *	31.3	11.8	28.8
RNN + GNN -> RNN	33.0	13.3	28.3
RNN -> RNN + pointer *	36.4	15.7	**33.4**
RNN + GNN -> RNN + pointer	**38.1**	**16.1**	33.2

** See et al. (2017). Get To The Point: Summarization with Pointer-Generator Networks*

NLP with GNNs

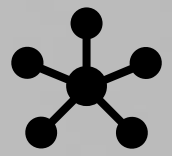
Grounded Language

Procedural Text



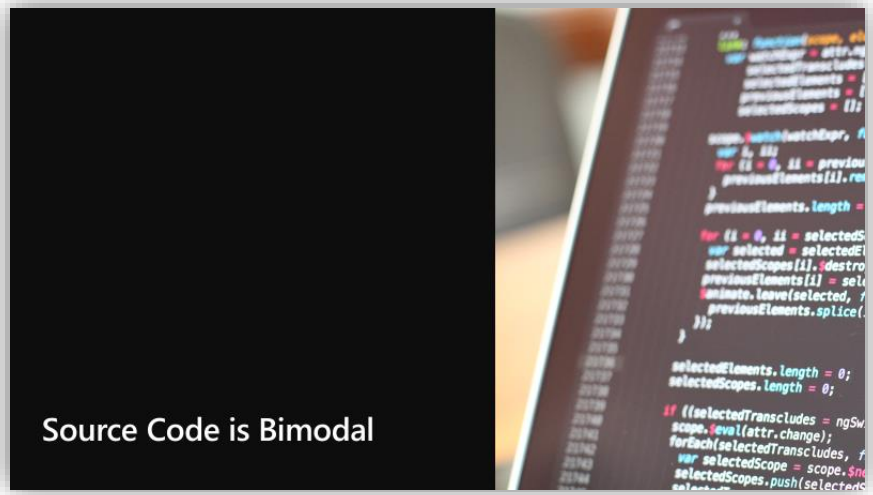
Semantic Parsing

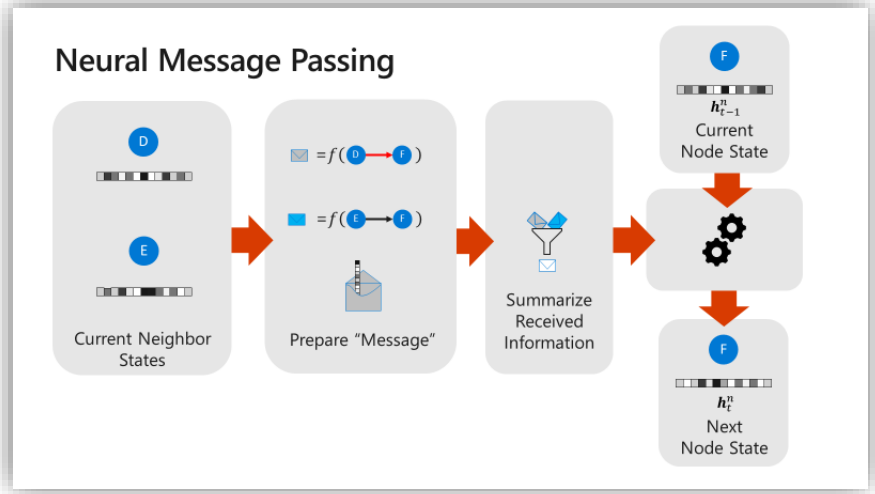
Source Code & Natural Language

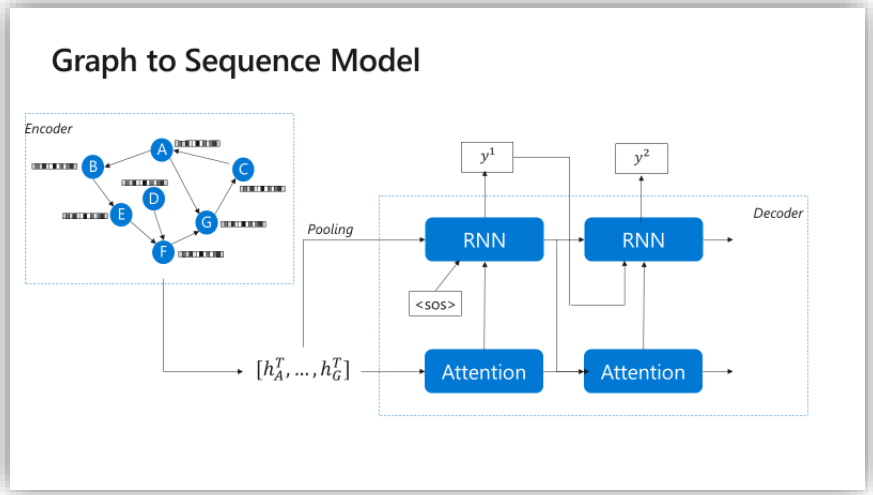


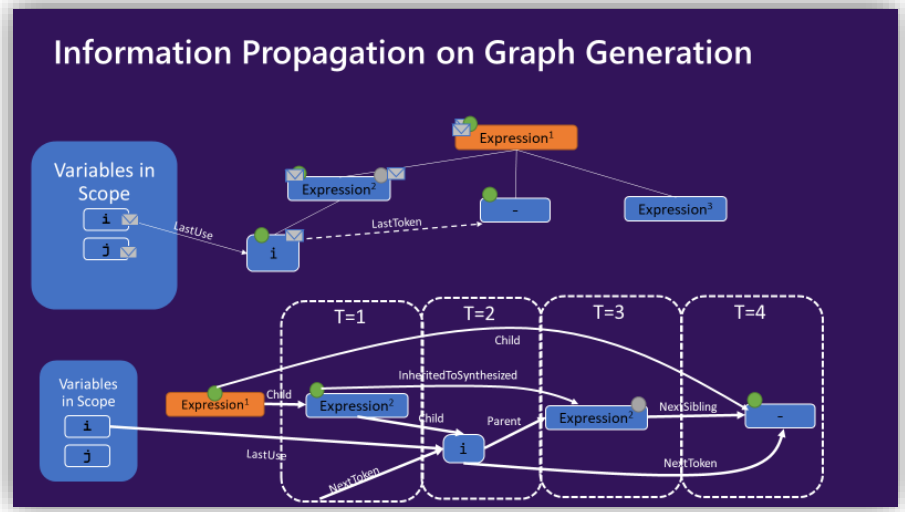
Rich Structure

Reasoning









Microsoft

Joint work with M. Brockschmidt, A.L. Gaunt, A. Polozov, P. Fernandes, M. Khademi