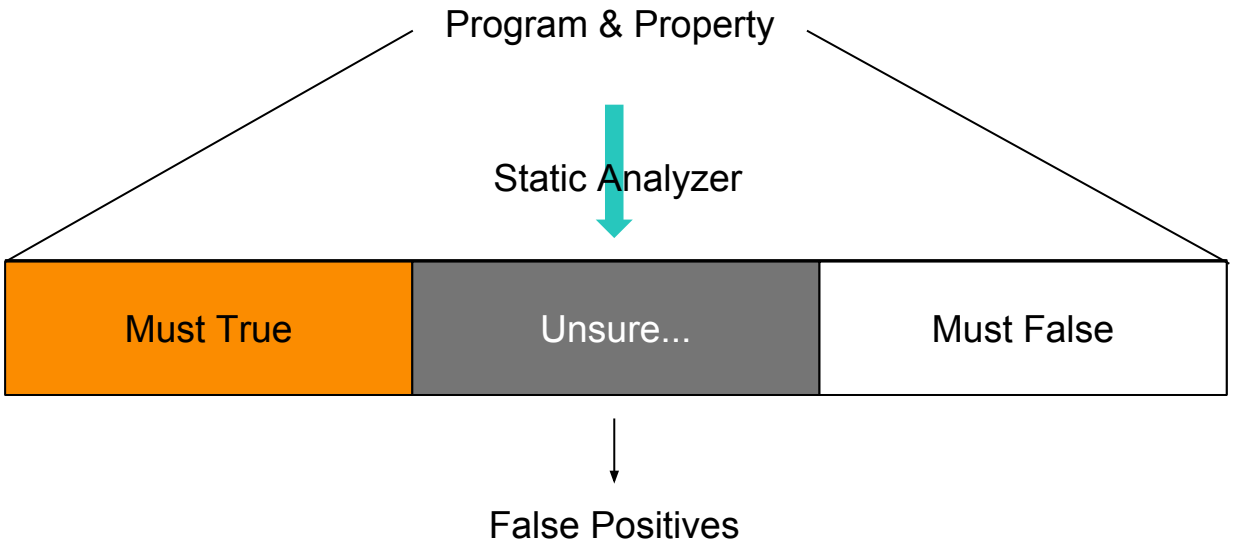
Neural-Augmented Static Analysis of Android Communication

Jinman Zhao, Aws Albarghouthi, Vaibhav Rastogi, Somesh Jha, Damien Octeau
University of Wisconsin-Madison, Google

Key Idea

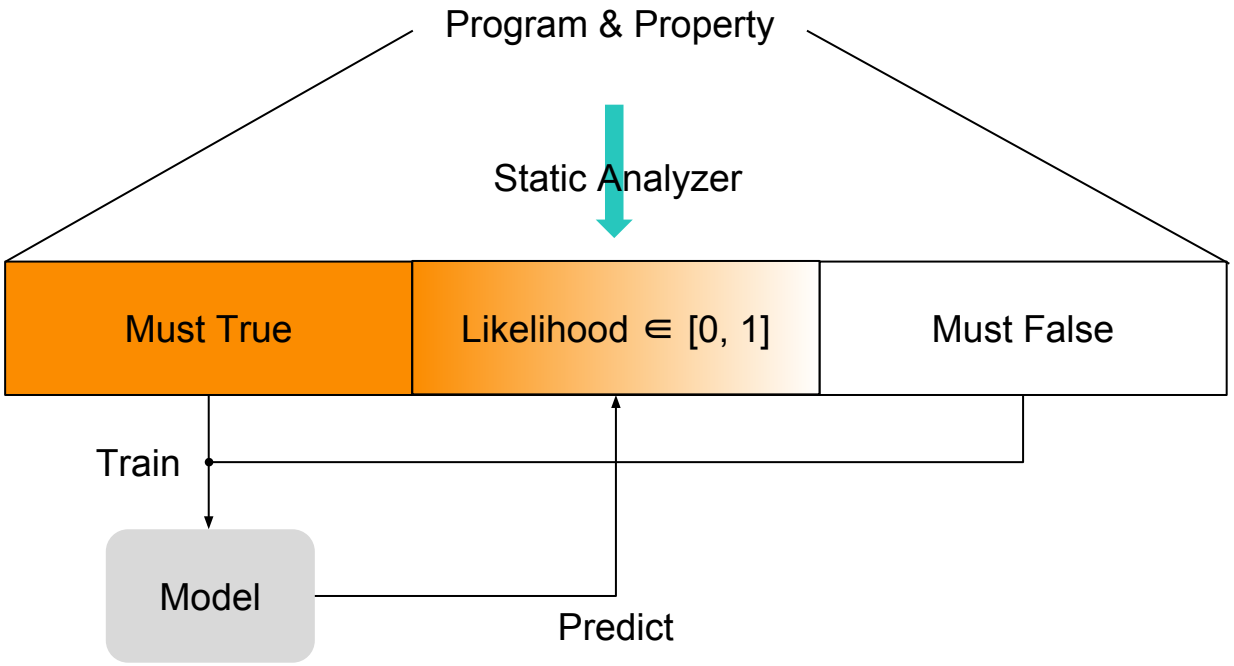
Use machine learning
to refine results
from static analysis.

Static Analysis: False Positives

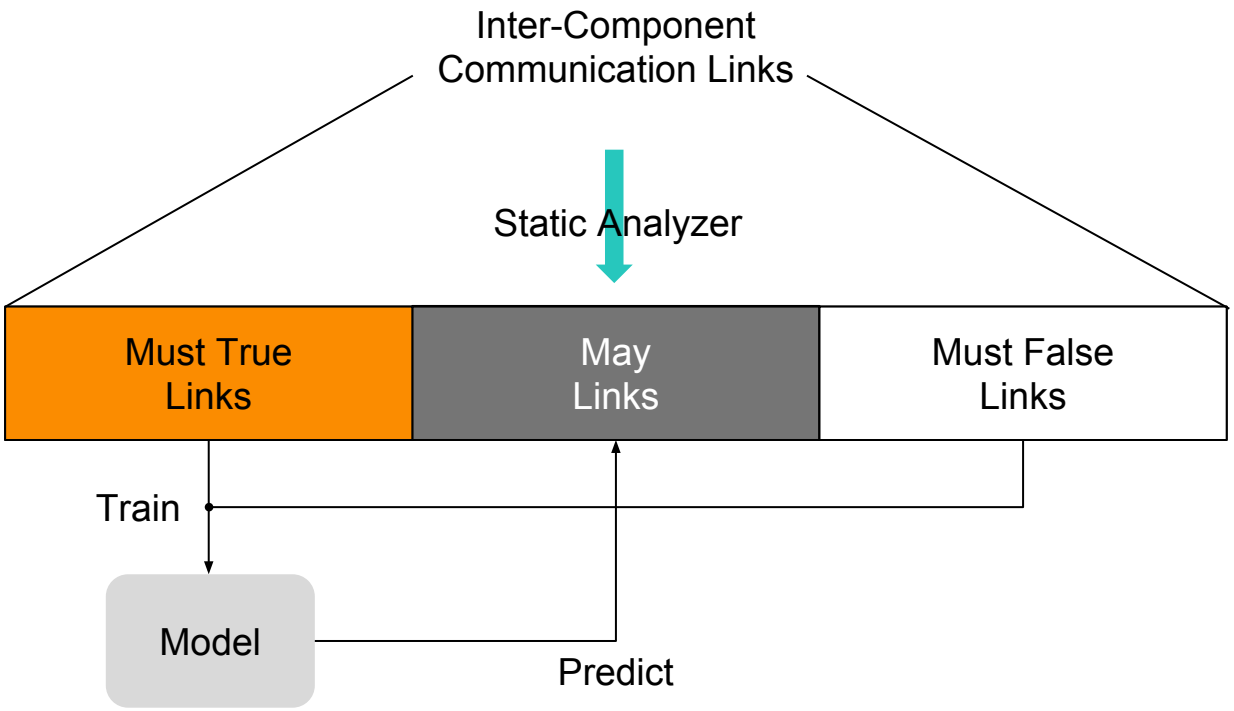


Ranking problem

Machine Learning to Augment



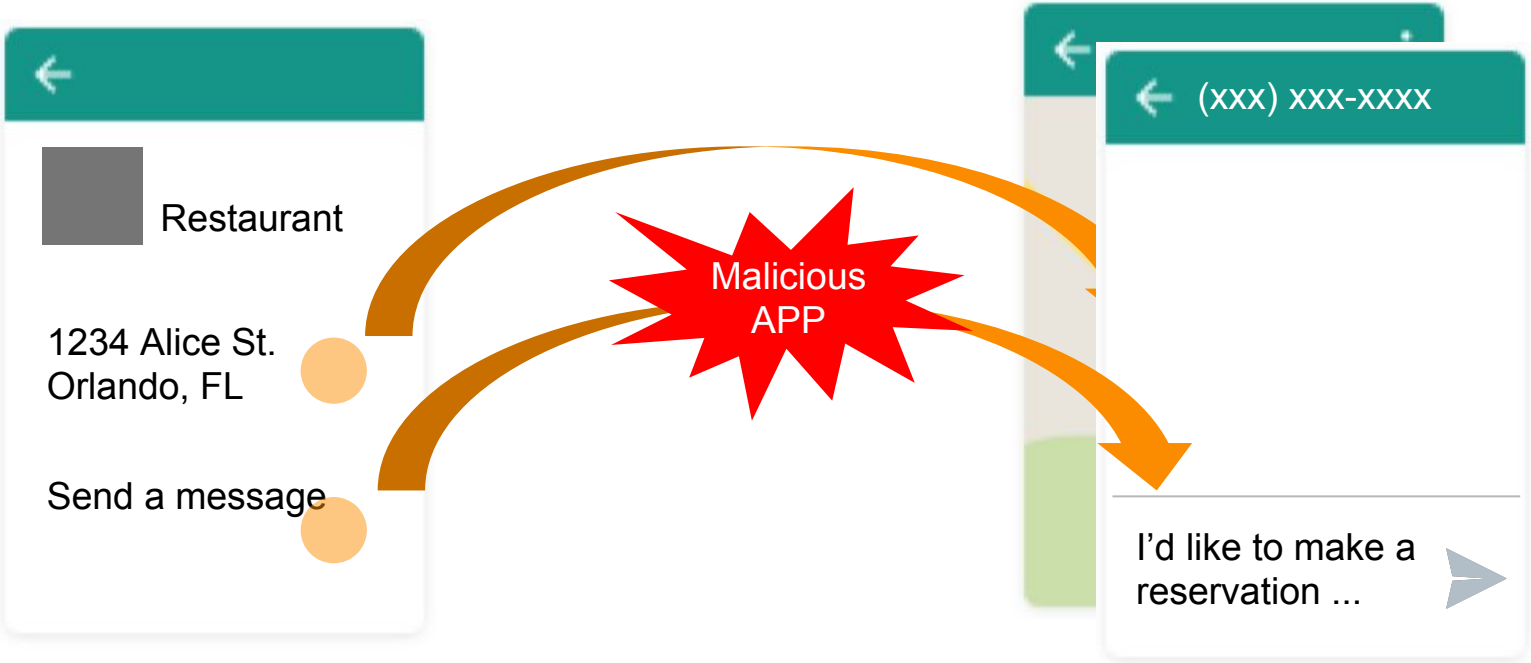
Link Inference for Android Communication



Task

Link Inference in
Android Communication

Android ICC: A User's Experience



Intent → *Inter-Component Communication* → Component w/ Filter

Android ICC: An Example

Code View

Intent

```
public void sendImplicitIntent() {
    Intent intent = new Intent();
    intent.setAction("SEND");
    msg = ... // contains phone # and msg
    intent.setData(msg);
    startActivity(intent);
}
```

Code constructing and starting implicit intent

Filter

```
<intent-filter>
    <action android:name="SEND"/>
    <action android:name="VIEW"/>
    <data android:scheme="sms"/>
    <category android:name="DEFAULT"/>
</intent-filter>
```

Intent filter for a sms component

(part of) the resolution logic

ICC link?

Yes!

$$match(i, f) = type(i, f) \wedge visibility(i, f) \wedge perm(i, f) \wedge (explicit(i, f) \vee implicit(i, f)).$$

$$type(i, f) = i_{type} \subseteq f_{type}$$

$$visibility(i, f) = i_{app\_name} \subseteq f_{app\_name} \vee f_{exported} \subseteq \{\texttt{true}\}$$

$$perm(i, f) = i_{perm} \subseteq f_{uses\_perm} \wedge f_{perm} \subseteq i_{uses\_perm}$$

$$explicit(i, f) = i_{target\_comp} \neq \varnothing \wedge i_{target\_app} \subseteq f_{app\_name} \wedge i_{target\_comp} \subseteq f_{comp\_name}$$

$$implicit(i, f) = i_{target\_comp} = \varnothing \wedge \boxed{i_{action} \subseteq f_{actions}} \wedge i_{category} \subseteq f_{categories} \wedge data(i, f),$$

(Bigger part of) the resolution logic

(Octeau et al., POPL'16)

Previous Work: PRIMO

- PRIMO (Octeau et al., POPL'16) uses a hand-crafted probabilistic model that assigns probabilities to ICC links inferred by static analysis.
  - Laborious, error-prone and requiring expert domain knowledge.
  - Difficulty catching up with constantly evolving Android system.

Questions

#1

How can we triage may links with minimal expert domain knowledge?

Neural networks.

#2

How can we process inputs of complex data types in a systematic way?

Type-directed encoder.

#3

How do our models perform?

Very good!

#4

Are the models learning the right things?

Seems like so.

We are not trying to…

- Propose new NN module
- Eliminate use of domain knowledge
- Rule out manual effort

We are trying to…

- Propose systematic way to construct NN
- Provide decent performance without expert knowledge
- Use less labour with more automation

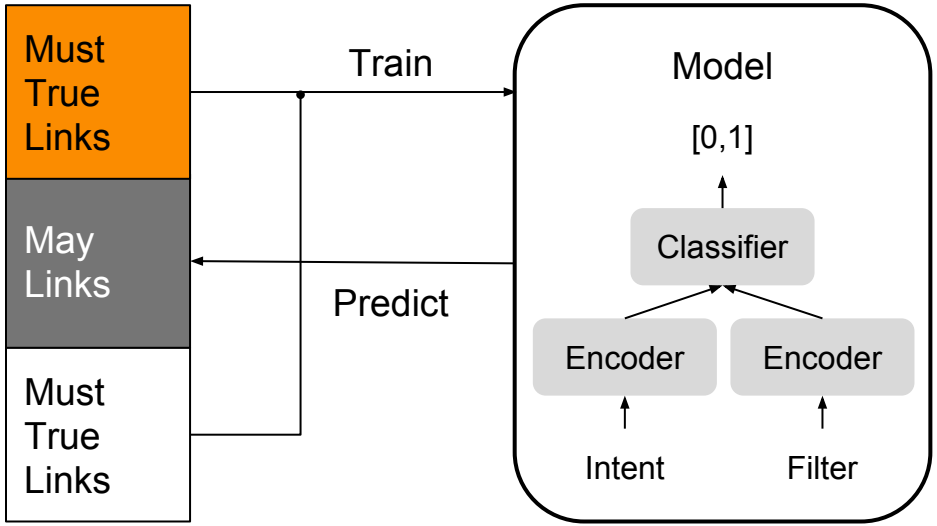
Approach

Part 1

How can we triage may links with minimal expert domain knowledge?

Link-Inference Neural Network

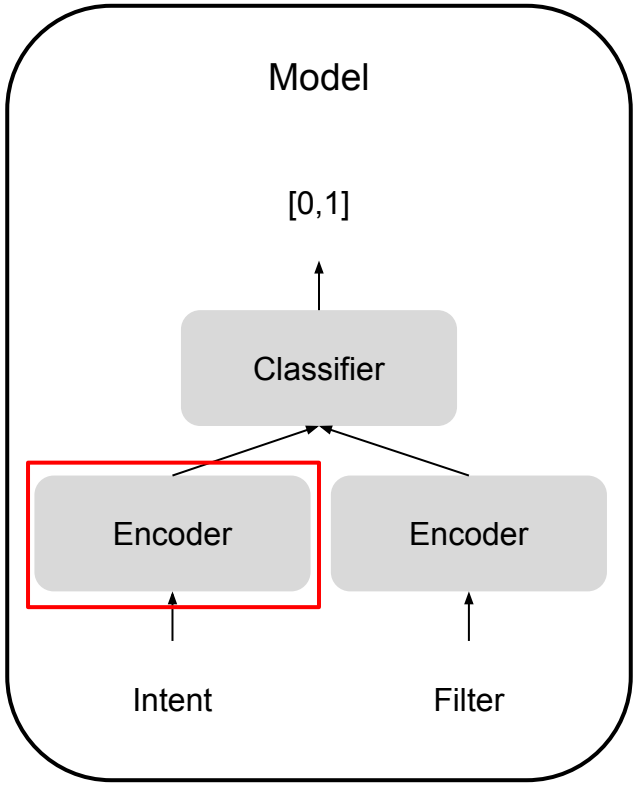
LINN: An end-to-end encoder-and-classifier architecture.



Approach

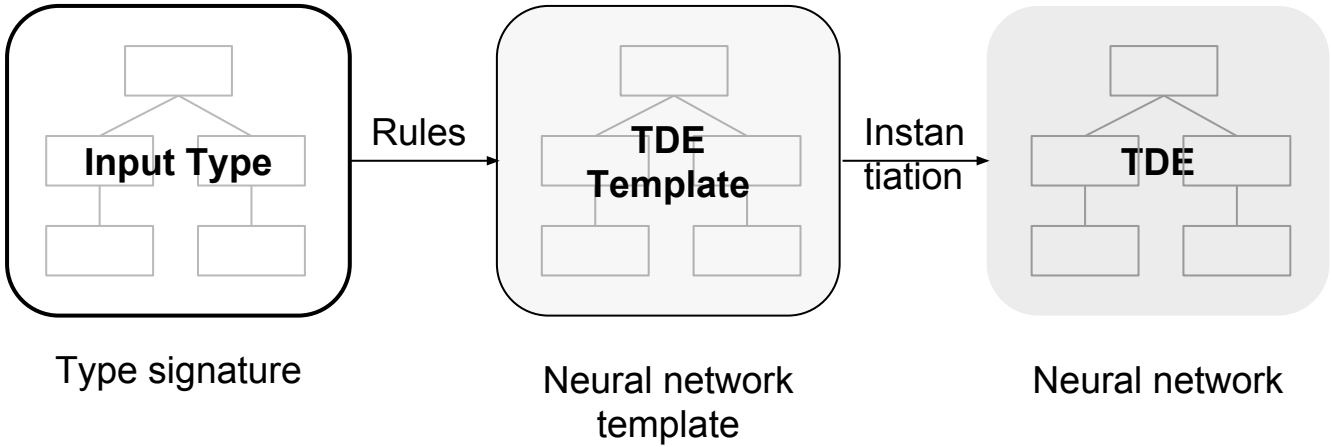
Part 2

How can we process inputs of complex data types in a systematic way?


Model
[0,1]
Classifier
Encoder
Encoder
Intent
Filter

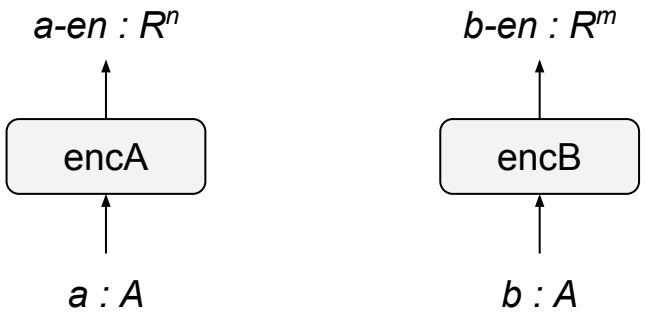
Type-Directed Encoder

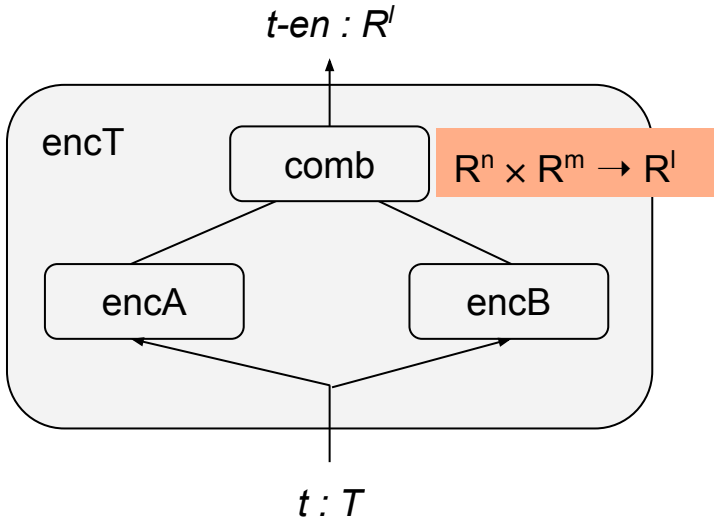
TDE: mapping type signature to neural network architecture.



An example: Encoding Product Types

Instance t := (a, b)
Type T := tuple(A, B)





$$\frac{g_1 : \tau_1 \rightarrow \mathbb{R}^n \quad g_2 : \tau_2 \rightarrow \mathbb{R}^m \quad comb : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^l}{\lambda(x, y).\, comb(g_1(x), g_2(y)) \blacktriangleright \tau_1 \times \tau_2} \text{ E-PROD}$$

$$\frac{}{\lambda x.\, x \blacktriangleright \mathbb{R}}\ \text{E-REAL}$$

$$\frac{enumEnc : \tau_c \to \mathbb{R}^n}{enumEnc \blacktriangleright \tau_c}\ \text{E-CAT}$$

$$\frac{\begin{array}{c} g : \tau \to \mathbb{R}^n \\ flat : \mathrm{L}(\mathbb{R}^n) \to \mathbb{R}^m \end{array}}{\lambda x.\, flat(map\ g\ x) \blacktriangleright \mathrm{L}(\tau)}\ \text{E-LIST}$$

$$\frac{\begin{array}{c} g : \tau \to \mathbb{R}^n \\ aggr : \mathrm{S}(\mathbb{R}^n) \to \mathbb{R}^m \end{array}}{\lambda x.\, aggr(map\ g\ x) \blacktriangleright \mathrm{S}(\tau)}\ \text{E-SET}$$

$$\boxed{\frac{g_1 : \tau_1 \to \mathbb{R}^n \quad g_2 : \tau_2 \to \mathbb{R}^m \quad comb : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^l}{\lambda(x, y).\, comb(g_1(x), g_2(y)) \blacktriangleright \tau_1 \times \tau_2}\ \text{E-PROD}}$$

$$\frac{g_1 : \tau_1 \to \mathbb{R}^n \quad g_2 : \tau_2 \to \mathbb{R}^n}{\lambda x.\, if\ x \in \tau_1\ then\ g_1(x)\ else\ g_2(x) \blacktriangleright \tau_1 + \tau_2}\ \text{E-SUM}$$

τ_c is a categorical type, e.g., characters.

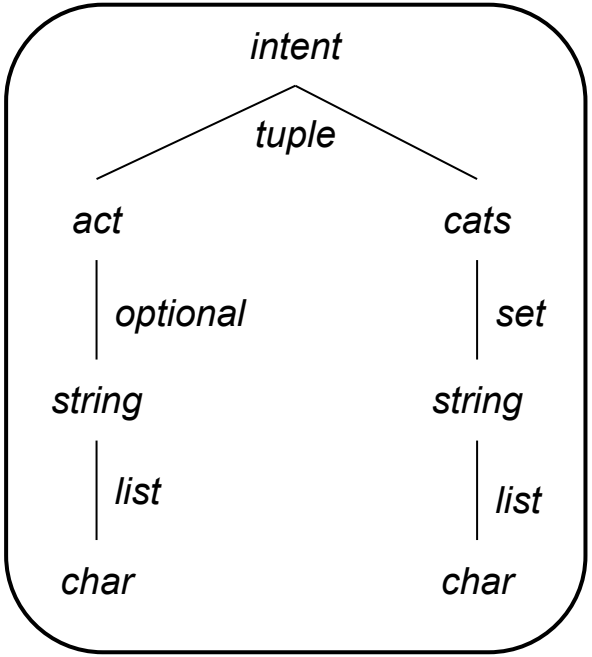
Rules for type-directed encoding

Android ICC: Our Abstraction

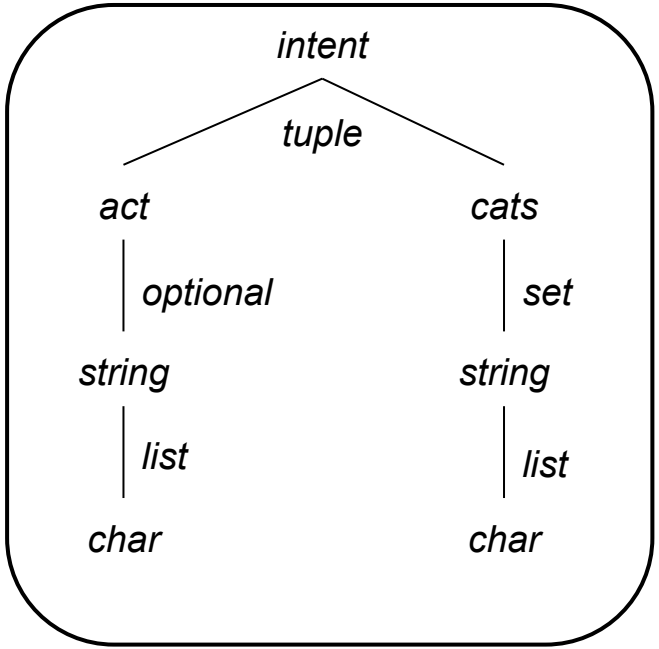
Type signatures

Intent *intent := tuple(act, cats)*
Action *act := optional(string)*
Categories *cats := set(string)*

Filter *filter := tuple(acts, cats)*
Actions *acts := set(string)*
Categories *cats := set(string)*

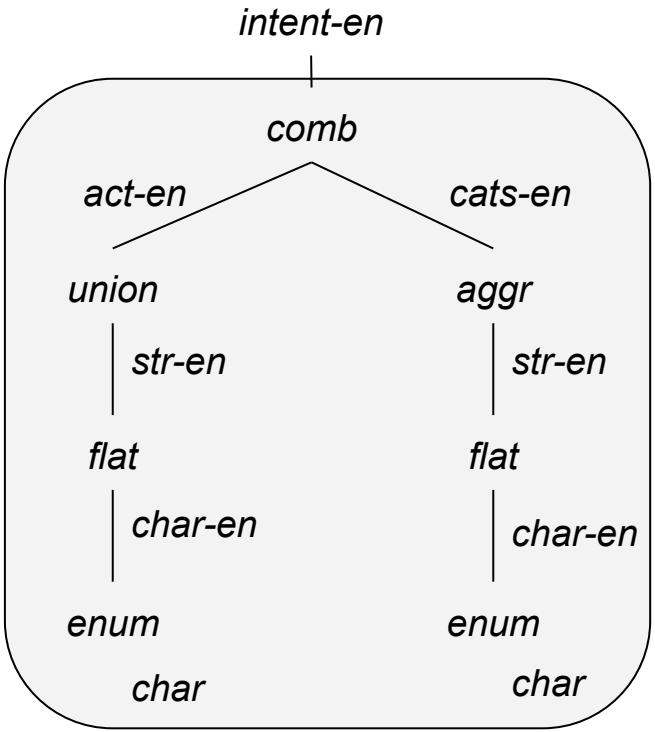


Type-Directed Encoder



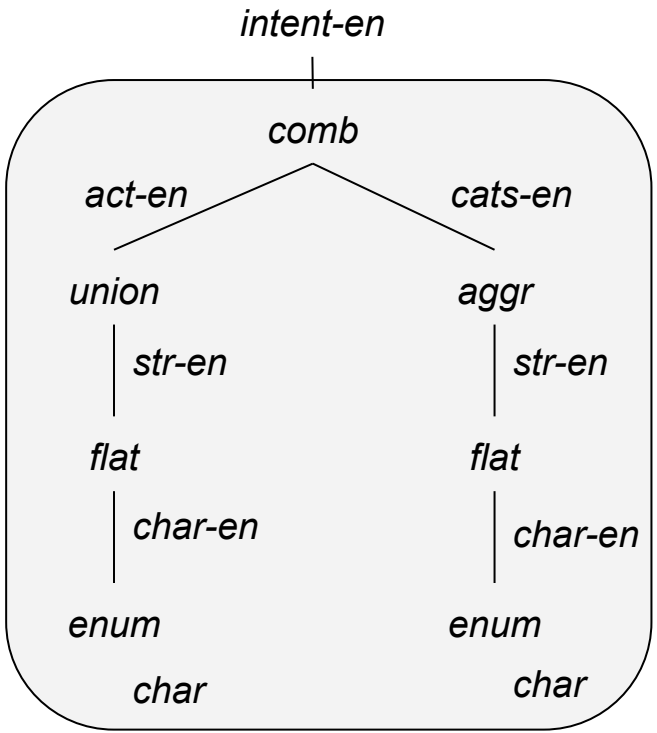
Type signature

Rules



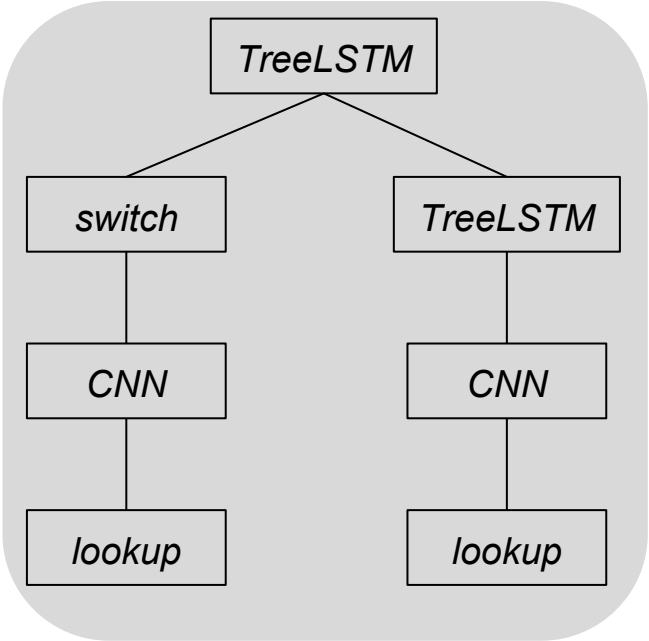
Neural network template

Type-Directed Encoder: Instantiation



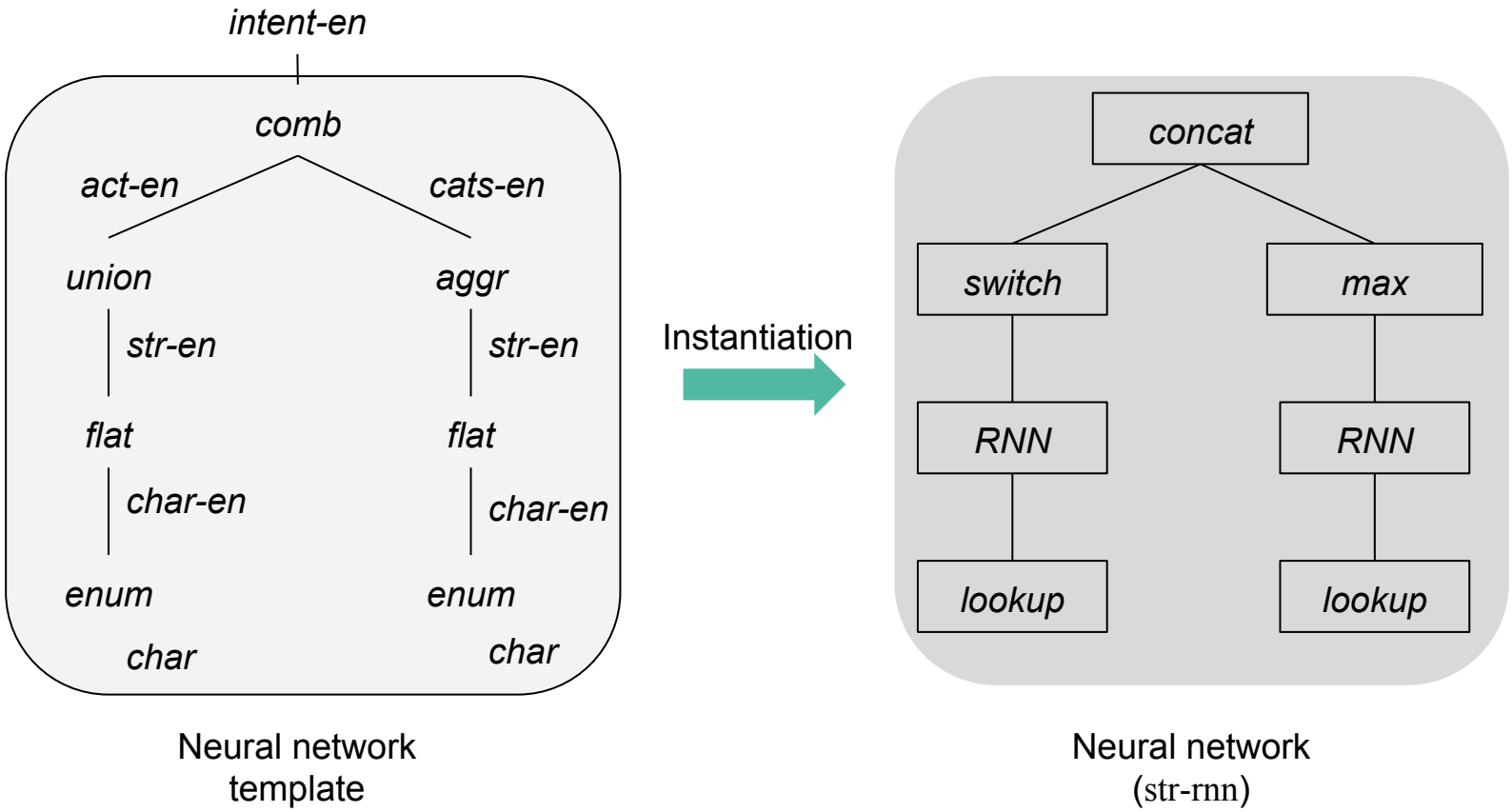
Neural network
template

Instantiation



Neural network
(typed-tree)

Type-Directed Encoder: Instantiation



A systematic way to build and explore structured NN.

Experiments

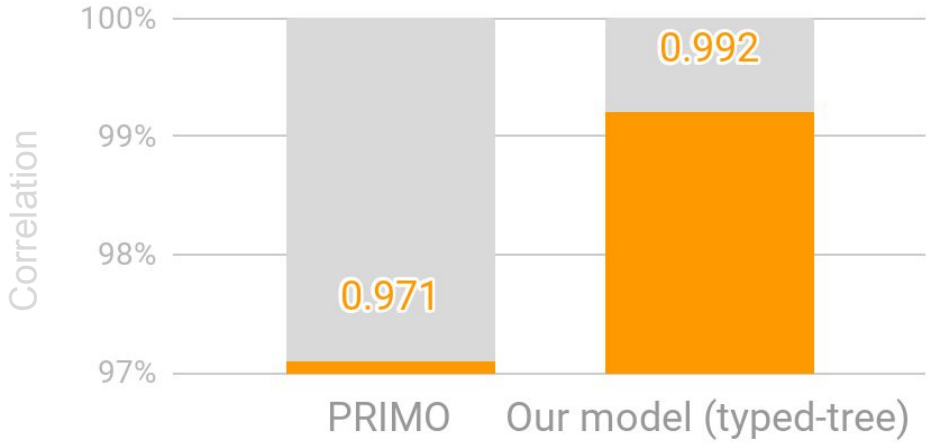
Are our models correctly predicting links?

Setup

- Dataset of 10,500 Android APPs from Google Play.
- IC3 (Octeau et al., ICSE'15) for static analysis.
- PRIMO's abstract matching for may/must partition.
- Simulated ground truth for may links.
- 4 instantiations of the TDE architecture.

	# pairs	# positive	# negative
training set	105,108	63,168	41,940
testing set	43,680	29,260	14,420

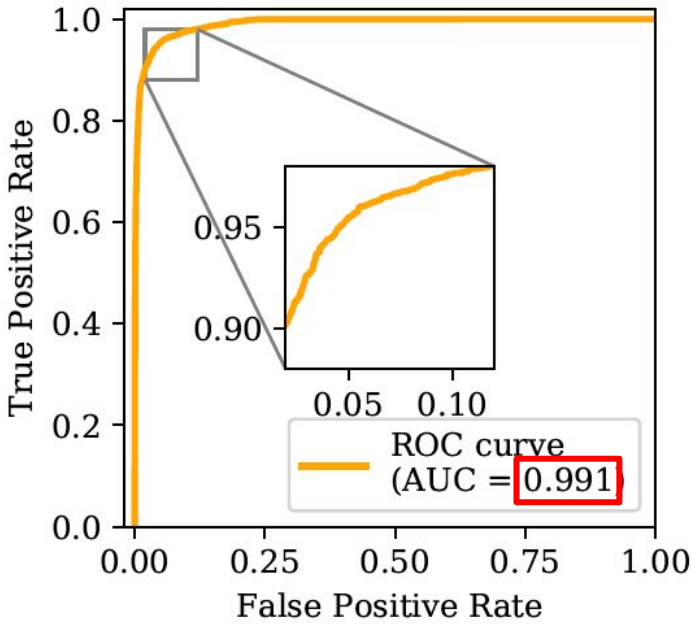
All instantiated models perform as good as PRIMO.

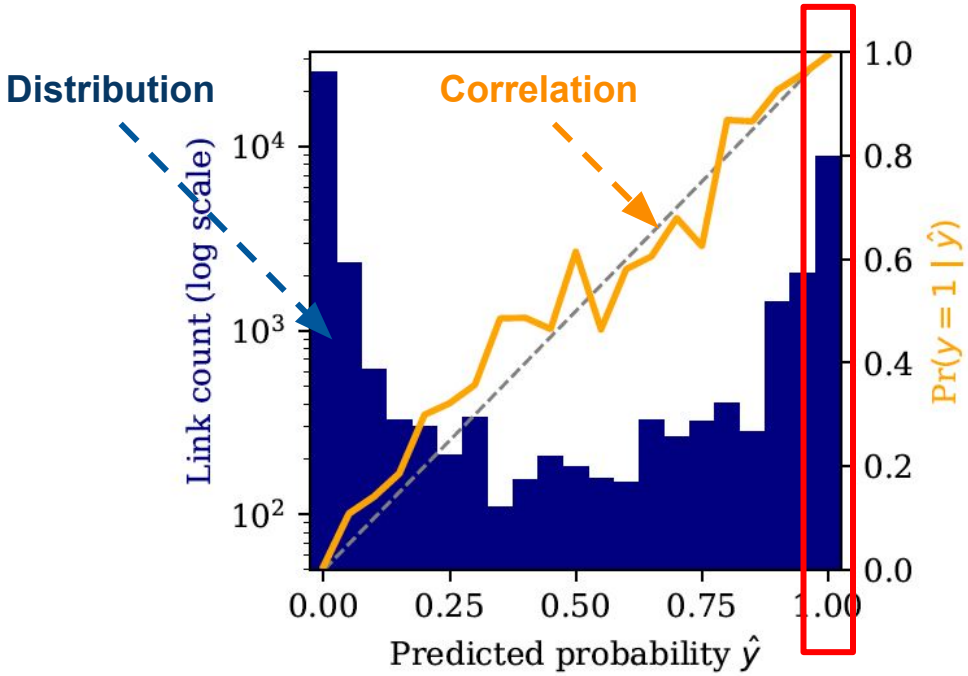


Our best model (typed-tree) fills the correlation gap by 72% compared to PRIMO despite the harder setting.

More Results for Our Best Model

ROC (left) and the distribution of predicted likelihood (right) from typed-tree model.





Interpretability

How do we know the model is learning the right thing?

Sensitivity to Masking

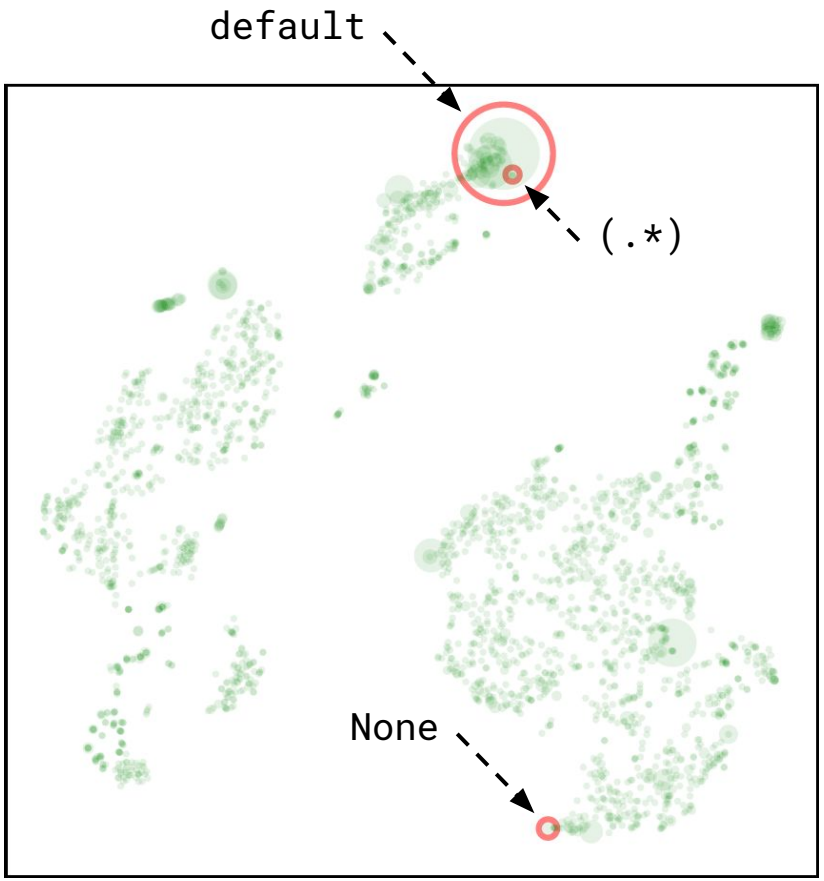
Picking distinctive values

Ignoring less useful parts


```
{"action": "NULL-CONSTANT", "categories": null}
{"actions": ["NULL-CONSTANTPOP_DIALOG", "NULL-CONSTANTPUSH_DIALOG_(.*)",
"(.*)REPLACE_DIALOG_(.*)", "APP-00489869YB964702HUPDATE_VIEW"], "categories":
null}

{"action": "NULL-CONSTANTREPLACE_DIALOG_(.*)", "categories": null}
{"actions": ["(.*).CLOSE"], "categories": null}

{"action": "(.*)", "categories": null}
{"actions": ["android.media.RINGER_MODE_CHANGED",
"sakurasoft.action.ALWAYS_LOCK", "android.intent.action.BOOT_COMPLETED"],
"categories": null}

{"action": "(.*)LOGIN_SUCCESS", "categories": null}
{"actions": ["NULL-CONSTANTLOGIN_FAIL", "NULL-
CONSTANTCREATE_PAYMENT_SUCCESS", "(.*)FATAL_ERROR",
"(.*)CREATE_PAYMENT_FAIL", "NULL-CONSTANTLOGIN_SUCCESS"], "categories": null}

{"action": "APP-00489869YB964702HREPLACE_DIALOG_(.*)", "categories": null}
{"actions": ["APP-00489869YB964702HLOGIN_FAIL", "APP-
00489869YB964702HCREATE_PAYMENT_FAIL", "NULL-CONSTANTCREATE_PAYMENT_SUCCESS",
"(.*)FATAL_ERROR", "NULL-CONSTANTLOGIN_SUCCESS"], "categories": null}

{"action": "com.joboevan.push.message.(.*)", "categories": null}
{"actions": ["com.joboevan.push.message.NULL-CONSTANT"], "categories": null}

{"action": "", "categories": ["(.*)"]}
{"actions": ["com.dreamware.Hells_Kitchen.CONCORRENTE"], "categories":
["android.intent.category.DEFAULT"]}

{"action": "(.*)", "categories": null}
{"actions": ["android.intent.action.MEDIA_BUTTON",
"com.ez.addon.MUSIC_COMMAND", "android.media.AUDIO_BECOMING_NOISY"],
"categories": null}
```

Learned Encodings

Semantically closer values receive more similar encodings.



Visualized by t-SNE.

Conclusion

- Neural-augmented static analysis
- Type-directed encoder
- Increased accuracy with less domain knowledge
- Interpretability study

Future Works

- Apply to other analysis tasks
- Push machine learning into static analysis procedure

Thanks for listening!
Q & A