DeepSTL –
From English Requirements to Signal Temporal Logic

Jie He, Ezio Bartocci, Dejan Ničković, Haris Isakovic, Radu Grosu
Cyber Physical Systems Group, TU Wien
AIT Austria Institute of Technology

April 13, 2022

Introduction

Alice: Verification Engineer

I spent my whole PhD studying formal methods.

However, I am frustrated by the resistance of my colleagues to use formal methods in practice.

- Difficult to learn
- **Hard to translate informal requirements into formal specifications!**

Bob: Machine Learning Expert

Why?

That’s True. But what does the specification look like?

Introduction

Alice: Verification Engineer

Take **Signal Temporal Logic (STL)** as an example.
It looks like this,

$$\varphi \quad := \quad x \sim u \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \mathbf{U}_I \varphi_2 \mid \varphi_1 \mathbf{S}_I \varphi_2$$

like this,

$$\begin{aligned}
(w,i) \models x \sim u \quad &\leftrightarrow \quad w(x,i) \sim u \\
(w,i) \models \neg\varphi \quad &\leftrightarrow \quad (w,i) \not\models \varphi \\
(w,i) \models \varphi_1 \vee \varphi_2 \quad &\leftrightarrow \quad (w,i) \models \varphi_1 \text{ or } (w,i) \models \varphi_2 \\
(w,i) \models \varphi_1 \mathbf{U}_I \varphi_2 \quad &\leftrightarrow \quad \exists j \in (i+I) \cap \mathbb{T} : (w,j) \models \varphi_2 \\
&\qquad \text{and } \forall i < k < j, (w,k) \models \varphi_1 \\
(w,i) \models \varphi_1 \mathbf{S}_I \varphi_2 \quad &\leftrightarrow \quad \exists j \in (i-I) \cap \mathbb{T} : (w,j) \models \varphi_2 \\
&\qquad \text{and } \forall j < k < i, (w,k) \models \varphi_1
\end{aligned}$$

Bob: Machine Learning Expert

Introduction

Alice: Verification Engineer

and this,

tautology	$\mathbf{true}$	$=$	$p \vee \neg p$
contradiction	$\mathbf{false}$	$=$	$\neg\mathbf{true}$
disjunction	$\varphi_1 \wedge \varphi_2$	$=$	$\neg(\neg\varphi_1 \vee \neg\varphi_2)$
implication	$\varphi_1 \rightarrow \varphi_2$	$=$	$\neg\varphi_1 \vee \varphi_2$
eventually, finally	$\mathbf{F}_I\varphi$	$=$	$\mathbf{true}\,\mathbf{U}_I\,\varphi$
always, globally	$\mathbf{G}_I\varphi$	$=$	$\neg\mathbf{F}_I\neg\varphi$
once	$\mathbf{O}_I\varphi$	$=$	$\mathbf{true}\,\mathbf{S}_I\,\varphi$
historically	$\mathbf{H}_I\varphi$	$=$	$\neg\mathbf{O}_I\neg\varphi$
rising edge	$\mathbf{rise}(\varphi)$	$=$	$\varphi \wedge \neg\varphi\,\mathbf{S}\,\mathbf{true}$
falling edge	$\mathbf{fall}(\varphi)$	$=$	$\neg\varphi \wedge \varphi\,\mathbf{S}\,\mathbf{true}$

Bob: Machine Learning Expert

A lot of math required!
I know why engineers are reluctant to use formal methods.

Introduction

Alice: Verification Engineer

Bob: Machine Learning Expert

Why not build a **translator** to directly translate English requirements into STL specifications?

Yes. Traditional approaches require strict rules for the input languages. But they are very flexible in reality.

Why not consider deep learning? Look at Google Translate, DeepL.

How?

Introduction

Architecture of DeepSTL

★ Most prominent contribution for deep-learning

Data Set

English Requirements → NL2STL Translator → Signal Temporal Logic (STL) Formula

AI Model

- **Highlights**

1. Empirical analysis of STL requirements found in scientific literature and practical applications that can guide data generation.

2. Generate synthetic examples of STL requirements that are consistent with the empirically collected statistics.

3. Employ a state-of-art machine translation architecture to train the translator, and evaluate its effectiveness of learning synthetic examples and its possibilities for extrapolation.

Introduction

Translation results of DeepSTL (Real output)

Requirement 1:

Whenever V_Mot is detected to become equal to 0, then at a time point starting after at most 100 time units Spd_Act shall continuously remain on 0 for at least 20 time units.

STL Formula:

```
always ( rise ( V_Mot == 0 ) -> eventually [ 0 : 100 ] ( always [ 0 : 20 ] ( Spd_Act == 0 ) ) )
```

Requirement 2:

Whenever the value of signal WEb gets changed to 40, then eventually Sd should equal to 5 at a certain moment during the next 1287 time units, till then the value of signal S4 must keep greater than or equal to 2755 and no more than 2771 all the time.

STL Formula:

```
always ( rise ( WEb == 40 ) -> ( S4 >= 2755 and S4 <= 2771 ) until [ 0 : 1287 ] ( Sd == 5 ) )
```

Introduction

Translation results of DeepSTL (Real output)

Requirement 3.1:
Whenever V_In is above 5, then there must exist a time point in the next 10 time units, at which the value of signal V_Out should be less than 2.
STL Formula:
```
always ( V_In > 5 -> eventually [ 0 : 10 ] ( V_Out < 2 ) )
```

Requirement 3.2:
Globally, if the value of V_In is greater than 5, then finally the value of V_Out should be smaller than 2 at a time point within 10 time units.
STL Formula:
```
always ( V_In > 5 -> eventually [ 0 : 10 ] ( V_Out < 2 ) )
```

Requirement 3.3:
It is always the case that when the signal V_In is larger than 5, then eventually at sometime during the following 10 time units the signal V_Out shall be smaller than 2.
STL Formula:
```
always ( V_In > 5 -> eventually [ 0 : 10 ] ( V_Out < 2 ) )
```

Step 1: Emipirical Analysis

Empirical Statistics – STL Specifications

1. STL Template Distribution

- Invariance/Reachability: $\mathbf{G}\varphi$, $\mathbf{G}_{[a\ b]}\varphi$, $\mathbf{F}_{[a\ b]}\varphi$

  Ex: $\mathbf{G}_{[\tau_s, T]}\ (\mu < c_l)$, $\mathbf{F}(x > 0.4)$

- Immediate response: $\mathbf{G}(\varphi \rightarrow \psi)$

  Ex: $\mathbf{G}(\text{not\_Eclipse} = 0 \rightarrow \text{sun\_currents} = 0)$

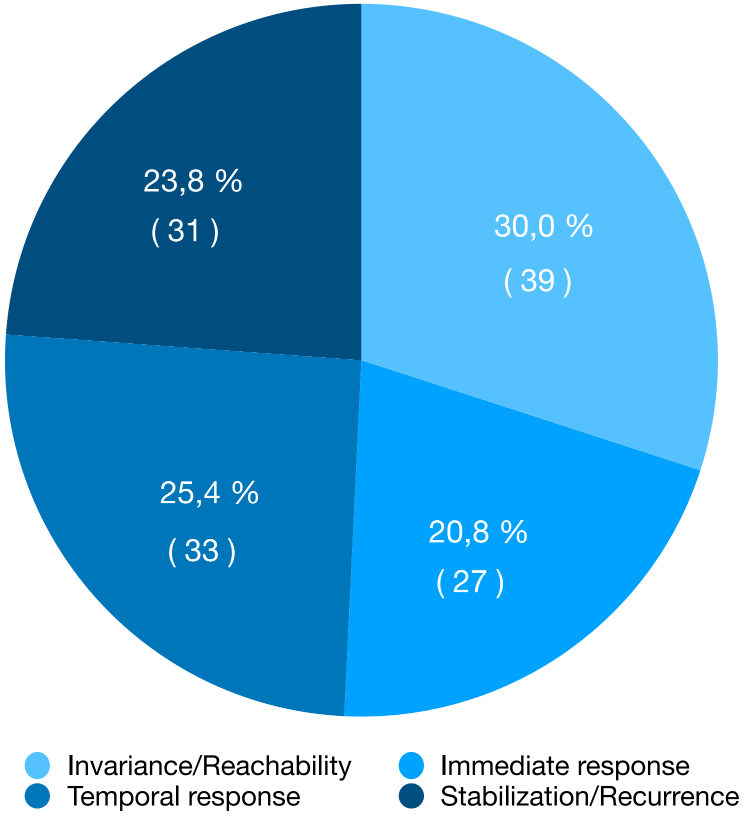
- Temporal response: $\mathbf{G}(\varphi' \rightarrow \psi')$

  Ex: $\mathbf{G}(\mathbf{rise}(\text{gear\_id} = 1) \rightarrow \mathbf{G}_{[0,2.5]}\, \neg\mathbf{fall}(\text{gear\_id} = 1))$

- Stabilization/Recurrence: $\mathbf{FG}\varphi$, $\mathbf{GF}\varphi$

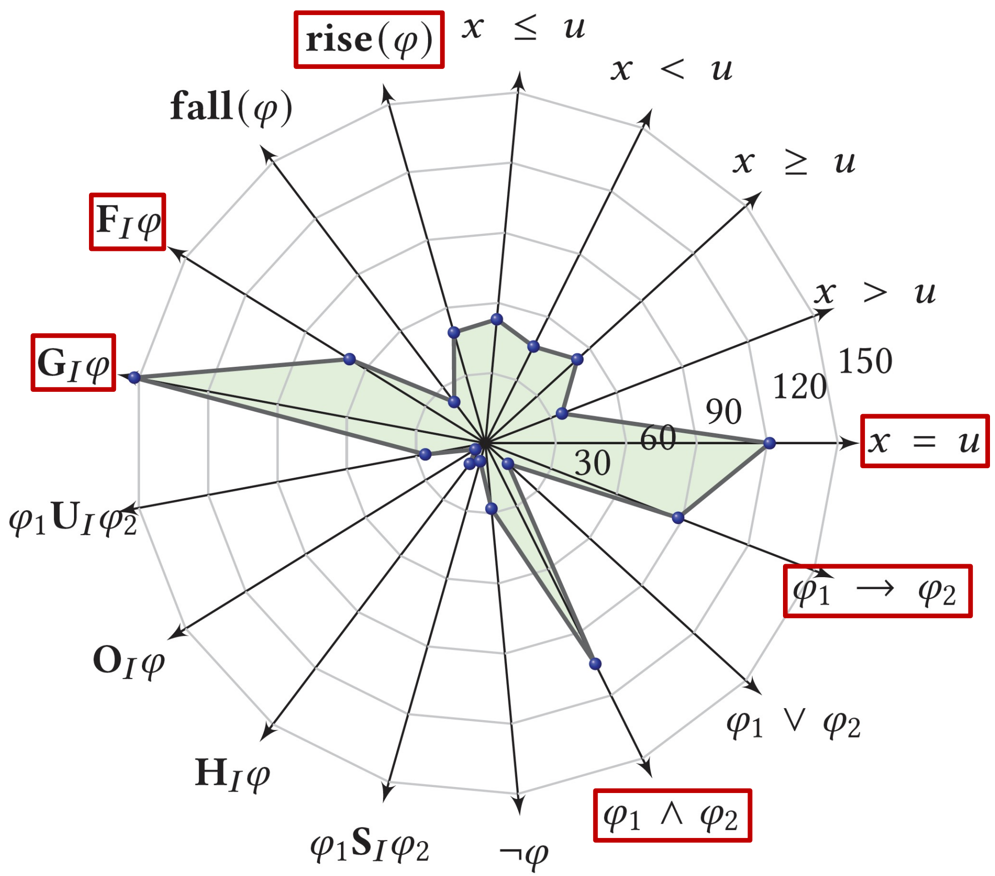
  Ex: $\mathbf{F}_{[0,\ 14400]}\mathbf{G}_{[4590,\ 9963]}\ (x_{10} \geq 0.325)$

  $\mathbf{G}_{[0,\ 12]}(\mathbf{F}_{[0,\ 2]}\ \text{regionA} \wedge \mathbf{F}_{[0,\ 2]}\ \text{regionB})$



Empirical Statistics – STL Specifications

2. STL Operator Distribution



- ❑ Equality ($x = u$):
  - Check if a discrete variable is in a given mode
- ❑ Conjunction ($\varphi_1 \wedge \varphi_2$):
  - Specify the value of a signal lying within a given range
- ❑ Implication ($\varphi_1 \rightarrow \varphi_2$):
  - Widely used in response specifications
- ❑ Rising edge ($\mathbf{rise}(\varphi)$):
  - A condition starts holding is more frequently specified
- ❑ **G** and **F** operator ($\mathbf{G}_I\varphi$, $\mathbf{F}_I\varphi$):
  - A majority of templates need to use them
- ❑ Future operator (**G**, **F**, **U**) vs. Past operator (**O**, **H**, **S**):
  - Most declarative specifications have a natural future flavor

Empirical Statistics – NL Requirements

1. English formulation of STL sentences

- Numeric (atomic) predicates
- Temporal operators (phrases)
- Specific scenarios (e.g., a rising/falling edge)

Sparsity and Imbalance

Example 1:

$x > \mu$	be above	increase above	be higher than	be larger than	be greater than	be bigger than	be more than	be over
Num.	4	2	1	1	1	0	0	0

Example 2:

$\mathbf{G}_{[0,t]} / \mathbf{H}_{[0,t]}$	for at least t time units	for more than t time units	for the following/past t time units	within t time units
Num.	8	6	0	0

Empirical Statistics – NL Requirements

2. Language Quality

- Clear (46 sentences, 35.4%)
  - Straightforward, unambiguous, without room for further interpretation
- Indirect (43 sentences, 33.1%)
  - Assume some implicit knowledge, need an expert for translation
- Ambiguous (41 sentences, 31.5%)
  - Vague, may have multiple interpretations
  - Lack key information from contexts
  - Rely on external sources, e.g., tables and figures

Step 2: Corpus Construction

Corpus Construction – Generate STL Formulas

1. Restricted STL Fragment

Simple Phrase (SP)

$$\mathrm{SP} \coloneqq \alpha \mid \alpha \wedge \alpha \mid \alpha \vee \alpha$$

$$\alpha \coloneqq x \circ u \mid \neg(x \circ u) \mid \textbf{rise}\ (x \circ u) \mid \textbf{fall}\ (x \circ u) \mid$$

$$\neg\textbf{rise}\ (x \circ u) \mid \neg\textbf{fall}\ (x \circ u)$$

$$\circ \in \{<, \leq, =, \geq, >\}$$

Corpus Construction – Generate STL Formulas

1. Restricted STL Fragment

Simple Phrase (SP)

$$\mathrm{SP} \coloneqq \alpha \mid \alpha \wedge \alpha \mid \alpha \vee \alpha$$

$$\alpha \coloneqq x \circ u \mid \neg(x \circ u) \mid \mathbf{rise}\ (x \circ u) \mid \mathbf{fall}\ (x \circ u) \mid \neg\mathbf{rise}\ (x \circ u) \mid \neg\mathbf{fall}\ (x \circ u)$$

$$\circ \in \{<, \leq, =, \geq, >\}$$

Temporal Phrase (TP)

$$\mathrm{TP} \coloneqq \mathrm{TP}' \mid \neg\mathrm{TP}' \mid \mathbf{rise}\ \mathrm{TP}' \mid \mathbf{fall}\ \mathrm{TP}' \mid \neg\mathbf{rise}\ \mathrm{TP}' \mid \neg\mathbf{fall}\ \mathrm{TP}'$$

$$\mathrm{TP}' \coloneqq \mathbf{UTO}_I(\alpha) \mid (\alpha)\mathbf{BTO}_I(\alpha)$$

$$\mathbf{UTO} \in \{\mathbf{F}, \mathbf{G}, \mathbf{O}, \mathbf{H}\},\ \mathbf{BTO} \in \{\mathbf{U}, \mathbf{S}\},\ I \in \{t_1, t_2\}$$

Corpus Construction – Generate STL Formulas

1. Restricted STL Fragment

Simple Phrase (SP)

$$\text{SP} \coloneqq \alpha \mid \alpha \wedge \alpha \mid \alpha \vee \alpha$$

$$\alpha \coloneqq x \circ u \mid \neg(x \circ u) \mid \mathbf{rise}\ (x \circ u) \mid \mathbf{fall}\ (x \circ u) \mid \neg\mathbf{rise}\ (x \circ u) \mid \neg\mathbf{fall}\ (x \circ u)$$

$$\circ \in \{<, \leq, =, \geq, >\}$$

Temporal Phrase (TP)

$$\text{TP} \coloneqq \text{TP}' \mid \neg\text{TP}' \mid \mathbf{rise}\ \text{TP}' \mid \mathbf{fall}\ \text{TP}' \mid \neg\mathbf{rise}\ \text{TP}' \mid \neg\mathbf{fall}\ \text{TP}'$$

$$\text{TP}' \coloneqq \mathbf{UTO}_I(\alpha) \mid (\alpha)\mathbf{BTO}_I(\alpha)$$

$$\mathbf{UTO} \in \{\mathbf{F}, \mathbf{G}, \mathbf{O}, \mathbf{H}\},\ \mathbf{BTO} \in \{\mathbf{U}, \mathbf{S}\},\ I \in \{t_1, t_2\}$$

Nested Temporal Phrase (NTP)

$$\text{NTP} \coloneqq \mathbf{F}_I\mathbf{G}_I(\alpha) \mid \mathbf{G}_I\mathbf{F}_I(\alpha)$$

Corpus Construction – Generate STL Formulas

1. Restricted STL Fragment

Simple Phrase (SP)

$$\mathrm{SP} \coloneqq \alpha \mid \alpha \wedge \alpha \mid \alpha \vee \alpha$$

$$\alpha \coloneqq x \circ u \mid \neg(x \circ u) \mid \textbf{rise } (x \circ u) \mid \textbf{fall } (x \circ u) \mid \neg\textbf{rise } (x \circ u) \mid \neg\textbf{fall } (x \circ u)$$

$$\circ \in \{<, \leq, =, \geq, >\}$$

Temporal Phrase (TP)

$$\mathrm{TP} \coloneqq \mathrm{TP}' \mid \neg\mathrm{TP}' \mid \textbf{rise } \mathrm{TP}' \mid \textbf{fall } \mathrm{TP}' \mid \neg\textbf{rise } \mathrm{TP}' \mid \neg\textbf{fall } \mathrm{TP}'$$

$$\mathrm{TP}' \coloneqq \mathbf{UTO}_I(\alpha) \mid (\alpha)\mathbf{BTO}_I(\alpha)$$

$$\mathbf{UTO} \in \{\mathbf{F}, \mathbf{G}, \mathbf{O}, \mathbf{H}\},\ \mathbf{BTO} \in \{\mathbf{U}, \mathbf{S}\},\ I \in \{t_1, t_2\}$$

Nested Temporal Phrase (NTP)

$$\mathrm{NTP} \coloneqq \mathbf{F}_I\mathbf{G}_I(\alpha) \mid \mathbf{G}_I\mathbf{F}_I(\alpha)$$

$$\mathrm{P} \coloneqq \mathrm{SP} \mid \mathrm{TP}$$

$$\psi \coloneqq \mathbf{G}_I\,(\mathrm{SP}) \mid \mathbf{F}_I(\mathrm{SP}) \quad \text{(Invariance/Reachability)}$$

$$\mid \mathbf{G}\,(\mathrm{SP} \rightarrow \mathrm{SP}) \quad \text{(Immediate response)}$$

$$\mid \mathbf{G}\,(\mathrm{P} \rightarrow \mathrm{TP}) \quad \text{(Temporal response)}$$

$$\mid \mathbf{G}\,(\mathrm{P} \rightarrow \mathrm{NTP}) \quad \text{(Stabilization/Recurrence)}$$

Corpus Construction – Generate STL Formulas

1. Restricted STL Fragment

Simple Phrase (SP)

$$\mathrm{SP} \coloneqq \alpha \mid \alpha \wedge \alpha \mid \alpha \vee \alpha$$

$$\alpha \coloneqq x \circ u \mid \neg(x \circ u) \mid \mathbf{rise}\ (x \circ u) \mid \mathbf{fall}\ (x \circ u) \mid \neg\mathbf{rise}\ (x \circ u) \mid \neg\mathbf{fall}\ (x \circ u)$$

$$\circ \in \{<, \leq, =, \geq, >\}$$

Temporal Phrase (TP)

$$\mathrm{TP} \coloneqq \mathrm{TP}' \mid \neg\mathrm{TP}' \mid \mathbf{rise}\ \mathrm{TP}' \mid \mathbf{fall}\ \mathrm{TP}' \mid \neg\mathbf{rise}\ \mathrm{TP}' \mid \neg\mathbf{fall}\ \mathrm{TP}'$$

$$\mathrm{TP}' \coloneqq \mathbf{UTO}_I(\alpha) \mid (\alpha)\mathbf{BTO}_I(\alpha)$$

$$\mathbf{UTO} \in \{\mathbf{F}, \mathbf{G}, \mathbf{O}, \mathbf{H}\},\ \mathbf{BTO} \in \{\mathbf{U}, \mathbf{S}\},\ I \in \{t_1, t_2\}$$

Nested Temporal Phrase (NTP)

$$\mathrm{NTP} \coloneqq \mathbf{F}_I\mathbf{G}_I(\alpha) \mid \mathbf{G}_I\mathbf{F}_I(\alpha)$$

$$\mathrm{P} \coloneqq \mathrm{SP} \mid \mathrm{TP}$$

$$\psi \coloneqq \mathbf{G}_I\,(\mathrm{SP}) \mid \mathbf{F}_I(\mathrm{SP}) \quad \text{(Invariance/Reachability)}$$
$$\mid \mathbf{G}\ (\mathrm{SP} \rightarrow \mathrm{SP}) \quad \text{(Immediate response)}$$
$$\mid \mathbf{G}\ (\mathrm{P} \rightarrow \mathrm{TP}) \quad \text{(Temporal response)}$$
$$\mid \mathbf{G}\ (\mathrm{P} \rightarrow \mathrm{NTP}) \quad \text{(Stabilization/Recurrence)}$$

Example (weighted sampling):

$$\mathbf{G}\ (\mathrm{V\_In} > 5 \rightarrow \mathbf{F}_{[0,\,10]}\ (\mathrm{V\_Out} < 2))$$

Corpus Construction – Generate STL Formulas

1. Restricted STL Fragment

Simple Phrase (SP)

$$\mathrm{SP} \coloneqq \alpha \mid \alpha \wedge \alpha \mid \alpha \vee \alpha$$

$$\alpha \coloneqq x \circ u \mid \neg(x \circ u) \mid \textbf{rise}\,(x \circ u) \mid \textbf{fall}\,(x \circ u) \mid \neg\textbf{rise}\,(x \circ u) \mid \neg\textbf{fall}\,(x \circ u)$$

$$\circ \in \{<, \leq, =, \geq, >\}$$

Temporal Phrase (TP)

$$\mathrm{TP} \coloneqq \mathrm{TP}' \mid \neg\mathrm{TP}' \mid \textbf{rise}\,\mathrm{TP}' \mid \textbf{fall}\,\mathrm{TP}' \mid \neg\textbf{rise}\,\mathrm{TP}' \mid \neg\textbf{fall}\,\mathrm{TP}'$$

$$\mathrm{TP}' \coloneqq \mathbf{UTO}_I(\alpha) \mid (\alpha)\mathbf{BTO}_I(\alpha)$$

$$\mathbf{UTO} \in \{\mathbf{F}, \mathbf{G}, \mathbf{O}, \mathbf{H}\},\ \mathbf{BTO} \in \{\mathbf{U}, \mathbf{S}\},\ I \in \{t_1, t_2\}$$

Nested Temporal Phrase (NTP)

$$\mathrm{NTP} \coloneqq \mathbf{F}_I\mathbf{G}_I(\alpha) \mid \mathbf{G}_I\mathbf{F}_I(\alpha)$$

$$\mathrm{P} \coloneqq \mathrm{SP} \mid \mathrm{TP}$$

$\psi \coloneqq$	$\mathbf{G}_I\,(\mathrm{SP}) \mid \mathbf{F}_I(\mathrm{SP})$	(Invariance/Reachability)
	$\mid \mathbf{G}\,(\mathrm{SP} \rightarrow \mathrm{SP})$	(Immediate response)
	$\mid \mathbf{G}\,(\mathrm{P} \rightarrow \mathrm{TP})$	(Temporal response)
	$\mid \mathbf{G}\,(\mathrm{P} \rightarrow \mathrm{NTP})$	(Stabilization/Recurrence)

Example (weighted sampling):

$$\mathbf{G}\,(\mathrm{V\_In} > 5 \rightarrow \mathbf{F}_{[0,\,10]}\ (\mathrm{V\_Out} < 2))$$

Corpus Construction – Generate STL Formulas

1. Restricted STL Fragment

Simple Phrase (SP)

$$\mathrm{SP} \coloneqq \alpha \mid \alpha \wedge \alpha \mid \alpha \vee \alpha$$

$$\alpha \coloneqq x \circ u \mid \neg(x \circ u) \mid \textbf{rise}\,(x \circ u) \mid \textbf{fall}\,(x \circ u) \mid \neg\textbf{rise}\,(x \circ u) \mid \neg\textbf{fall}\,(x \circ u)$$

$$\circ \in \{<, \leq, =, \geq, >\}$$

Temporal Phrase (TP)

$$\mathrm{TP} \coloneqq \mathrm{TP}' \mid \neg\mathrm{TP}' \mid \textbf{rise}\,\mathrm{TP}' \mid \textbf{fall}\,\mathrm{TP}' \mid \neg\textbf{rise}\,\mathrm{TP}' \mid \neg\textbf{fall}\,\mathrm{TP}'$$

$$\mathrm{TP}' \coloneqq \mathbf{UTO}_I(\alpha) \mid (\alpha)\mathbf{BTO}_I(\alpha)$$

$$\mathbf{UTO} \in \{\mathbf{F}, \mathbf{G}, \mathbf{O}, \mathbf{H}\},\ \mathbf{BTO} \in \{\mathbf{U}, \mathbf{S}\},\ I \in \{t_1, t_2\}$$

Nested Temporal Phrase (NTP)

$$\mathrm{NTP} \coloneqq \mathbf{F}_I\mathbf{G}_I(\alpha) \mid \mathbf{G}_I\mathbf{F}_I(\alpha)$$

$$\mathrm{P} \coloneqq \mathrm{SP} \mid \mathrm{TP}$$

$\psi \coloneqq$	$\mathbf{G}_I\,(\mathrm{SP}) \mid \mathbf{F}_I(\mathrm{SP})$	(Invariance/Reachability)
	$\mid \mathbf{G}\,(\mathrm{SP} \rightarrow \mathrm{SP})$	(Immediate response)
	$\mid \mathbf{G}\,(\mathrm{P} \rightarrow \mathrm{TP})$	(Temporal response)
	$\mid \mathbf{G}\,(\mathrm{P} \rightarrow \mathrm{NTP})$	(Stabilization/Recurrence)

Example (weighted sampling):

$$\mathbf{G}\,(\mathrm{V\_In} > 5 \rightarrow \mathbf{F}_{[0,\,10]}\ (\mathrm{V\_Out} < 2))$$

Corpus Construction – Generate STL Formulas

1. Restricted STL Fragment

Simple Phrase (SP)

$$\mathrm{SP} \coloneqq \alpha \mid \alpha \wedge \alpha \mid \alpha \vee \alpha$$

$$\alpha \coloneqq x \circ u \mid \neg(x \circ u) \mid \textbf{rise}\,(x \circ u) \mid \textbf{fall}\,(x \circ u) \mid \neg\textbf{rise}\,(x \circ u) \mid \neg\textbf{fall}\,(x \circ u)$$

$$\circ \in \{<, \leq, =, \geq, >\}$$

Temporal Phrase (TP)

$$\mathrm{TP} \coloneqq \mathrm{TP}' \mid \neg\mathrm{TP}' \mid \textbf{rise}\,\mathrm{TP}' \mid \textbf{fall}\,\mathrm{TP}' \mid \neg\textbf{rise}\,\mathrm{TP}' \mid \neg\textbf{fall}\,\mathrm{TP}'$$

$$\mathrm{TP}' \coloneqq \mathbf{UTO}_I(\alpha) \mid (\alpha)\mathbf{BTO}_I(\alpha)$$

$$\mathbf{UTO} \in \{\mathbf{F}, \mathbf{G}, \mathbf{O}, \mathbf{H}\},\ \mathbf{BTO} \in \{\mathbf{U}, \mathbf{S}\},\ I \in \{t_1, t_2\}$$

Nested Temporal Phrase (NTP)

$$\mathrm{NTP} \coloneqq \mathbf{F}_I\mathbf{G}_I(\alpha) \mid \mathbf{G}_I\mathbf{F}_I(\alpha)$$

$$\mathrm{P} \coloneqq \mathrm{SP} \mid \mathrm{TP}$$

$$\begin{aligned}\psi \coloneqq\ & \mathbf{G}_I\,(\mathrm{SP}) \mid \mathbf{F}_I(\mathrm{SP}) && \text{(Invariance/Reachability)}\\ & \mid \mathbf{G}\,(\mathrm{SP} \rightarrow \mathrm{SP}) && \text{(Immediate response)}\\ & \mid \mathbf{G}\,(\mathrm{P} \rightarrow \mathrm{TP}) && \text{(Temporal response)}\\ & \mid \mathbf{G}\,(\mathrm{P} \rightarrow \mathrm{NTP}) && \text{(Stabilization/Recurrence)}\end{aligned}$$

Example (weighted sampling):

$$\mathbf{G}\,(\mathrm{V\_In} > 5 \rightarrow \mathbf{F}_{[0,\,10]}\ (\mathrm{V\_Out} < 2))$$

Corpus Construction – Generate STL Formulas

1. Restricted STL Fragment

Simple Phrase (SP)

$SP \coloneqq \alpha \mid \alpha \wedge \alpha \mid \alpha \vee \alpha$

$\alpha \coloneqq x \circ u \mid \neg(x \circ u) \mid \textbf{rise}\ (x \circ u) \mid \textbf{fall}\ (x \circ u) \mid \neg\textbf{rise}\ (x \circ u) \mid \neg\textbf{fall}\ (x \circ u)$

$\circ \in \{<, \leq, =, \geq, >\}$

Temporal Phrase (TP)

$TP \coloneqq TP' \mid \neg TP' \mid \textbf{rise}\ TP' \mid \textbf{fall}\ TP' \mid \neg\textbf{rise}\ TP' \mid \neg\textbf{fall}\ TP'$

$TP' \coloneqq \textbf{UTO}_I(\alpha) \mid (\alpha)\textbf{BTO}_I(\alpha)$

$\textbf{UTO} \in \{\textbf{F}, \textbf{G}, \textbf{O}, \textbf{H}\}$, $\textbf{BTO} \in \{\textbf{U}, \textbf{S}\}$, $I \in \{t_1, t_2\}$

Nested Temporal Phrase (NTP)

$NTP \coloneqq \textbf{F}_I\textbf{G}_I(\alpha) \mid \textbf{G}_I\textbf{F}_I(\alpha)$

$P \coloneqq SP \mid TP$

$\psi \coloneqq \textbf{G}_I\,(SP) \mid \textbf{F}_I(SP)$ (Invariance/Reachability)

$\mid \textbf{G}\ (SP \rightarrow SP)$ (Immediate response)

$\mid \textbf{G}\ (P \rightarrow TP)$ (Temporal response)

$\mid \textbf{G}\ (P \rightarrow NTP)$ (Stabilization/Recurrence)

Example (weighted sampling):

$\textbf{G}\ (V\_In > 5 \rightarrow \textbf{F}_{[0,\,10]}\ (V\_Out < 2))$

Corpus Construction – Generate STL Formulas

1. Restricted STL Fragment

Simple Phrase (SP)

$\mathrm{SP} \coloneqq \alpha \mid \alpha \wedge \alpha \mid \alpha \vee \alpha$

$\alpha \coloneqq x \circ u \mid \neg(x \circ u) \mid \mathbf{rise}\ (x \circ u) \mid \mathbf{fall}\ (x \circ u) \mid \neg\mathbf{rise}\ (x \circ u) \mid \neg\mathbf{fall}\ (x \circ u)$

$\circ \in \{<, \leq, =, \geq, >\}$

Temporal Phrase (TP)

$\mathrm{TP} \coloneqq \mathrm{TP}' \mid \neg\mathrm{TP}' \mid \mathbf{rise}\ \mathrm{TP}' \mid \mathbf{fall}\ \mathrm{TP}' \mid \neg\mathbf{rise}\ \mathrm{TP}' \mid \neg\mathbf{fall}\ \mathrm{TP}'$

$\mathrm{TP}' \coloneqq \mathbf{UTO}_I(\alpha) \mid (\alpha)\mathbf{BTO}_I(\alpha)$

$\mathbf{UTO} \in \{\mathbf{F}, \mathbf{G}, \mathbf{O}, \mathbf{H}\}$, $\mathbf{BTO} \in \{\mathbf{U}, \mathbf{S}\}$, $I \in \{t_1, t_2\}$

Nested Temporal Phrase (NTP)

$\mathrm{NTP} \coloneqq \mathbf{F}_I\mathbf{G}_I(\alpha) \mid \mathbf{G}_I\mathbf{F}_I(\alpha)$

$\mathrm{P} \coloneqq \mathrm{SP} \mid \mathrm{TP}$

$\psi \coloneqq \mathbf{G}_I\,(\mathrm{SP}) \mid \mathbf{F}_I(\mathrm{SP})$	(Invariance/Reachability)
$\mid \mathbf{G}\ (\mathrm{SP} \rightarrow \mathrm{SP})$	(Immediate response)
$\mid \mathbf{G}\ (\mathrm{P} \rightarrow \mathrm{TP})$	(Temporal response)
$\mid \mathbf{G}\ (\mathrm{P} \rightarrow \mathrm{NTP})$	(Stabilization/Recurrence)

Example (weighted sampling):

$\mathbf{G}\ (\mathrm{V\_In} > 5 \rightarrow \mathbf{F}_{[0,\,10]}\ (\mathrm{V\_Out} < 2))$

Corpus Construction – Generate STL Formulas

1. Restricted STL Fragment

Simple Phrase (SP)

$$\mathrm{SP} \coloneqq \alpha \mid \alpha \wedge \alpha \mid \alpha \vee \alpha$$

$$\alpha \coloneqq x \circ u \mid \neg(x \circ u) \mid \textbf{rise}\,(x \circ u) \mid \textbf{fall}\,(x \circ u) \mid \neg\textbf{rise}\,(x \circ u) \mid \neg\textbf{fall}\,(x \circ u)$$

$$\circ \in \{<, \leq, =, \geq, >\}$$

Temporal Phrase (TP)

$$\mathrm{TP} \coloneqq \mathrm{TP}' \mid \neg\mathrm{TP}' \mid \textbf{rise}\ \mathrm{TP}' \mid \textbf{fall}\ \mathrm{TP}' \mid \neg\textbf{rise}\ \mathrm{TP}' \mid \neg\textbf{fall}\ \mathrm{TP}'$$

$$\mathrm{TP}' \coloneqq \mathbf{UTO}_I(\alpha) \mid (\alpha)\mathbf{BTO}_I(\alpha)$$

$$\mathbf{UTO} \in \{\mathbf{F}, \mathbf{G}, \mathbf{O}, \mathbf{H}\},\ \mathbf{BTO} \in \{\mathbf{U}, \mathbf{S}\},\ I \in \{t_1, t_2\}$$

Nested Temporal Phrase (NTP)

$$\mathrm{NTP} \coloneqq \mathbf{F}_I\mathbf{G}_I(\alpha) \mid \mathbf{G}_I\mathbf{F}_I(\alpha)$$

$$\mathrm{P} \coloneqq \mathrm{SP} \mid \mathrm{TP}$$

$\psi \coloneqq \mathbf{G}_I\,(\mathrm{SP}) \mid \mathbf{F}_I(\mathrm{SP})$	(Invariance/Reachability)
$\mid \mathbf{G}\,(\mathrm{SP} \rightarrow \mathrm{SP})$	(Immediate response)
$\mid \mathbf{G}\,(\mathrm{P} \rightarrow \mathrm{TP})$	(Temporal response)
$\mid \mathbf{G}\,(\mathrm{P} \rightarrow \mathrm{NTP})$	(Stabilization/Recurrence)

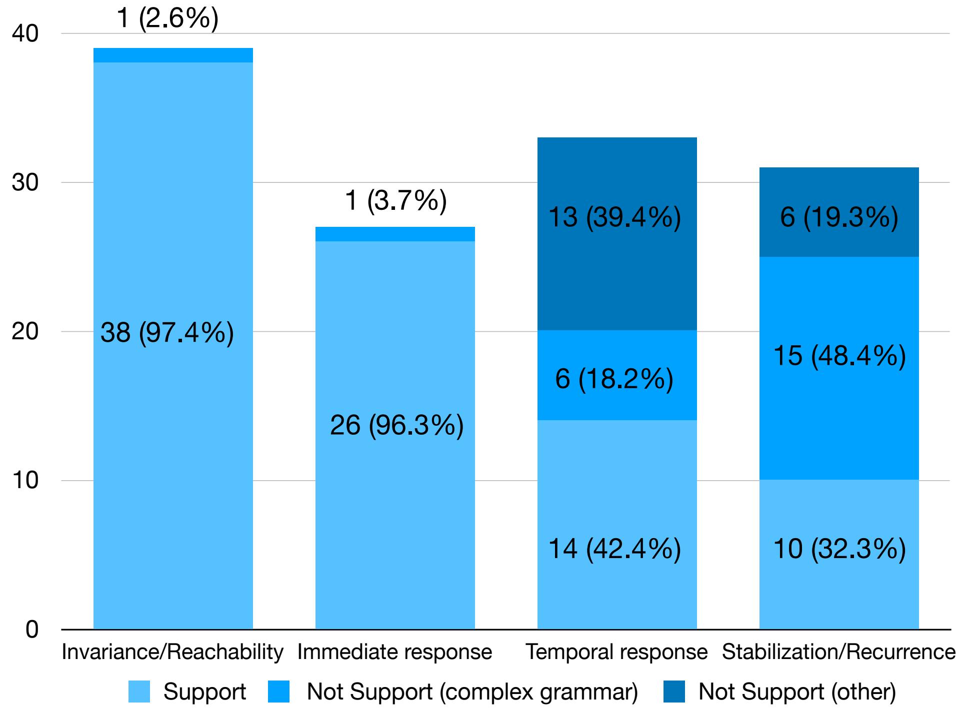
Example (weighted sampling):

$$\mathbf{G}\,(\mathrm{V\_In} > 5 \rightarrow \mathbf{F}_{[0,\,10]}\,(\mathrm{V\_Out} < 2))$$

Corpus Construction – Generate STL Formulas

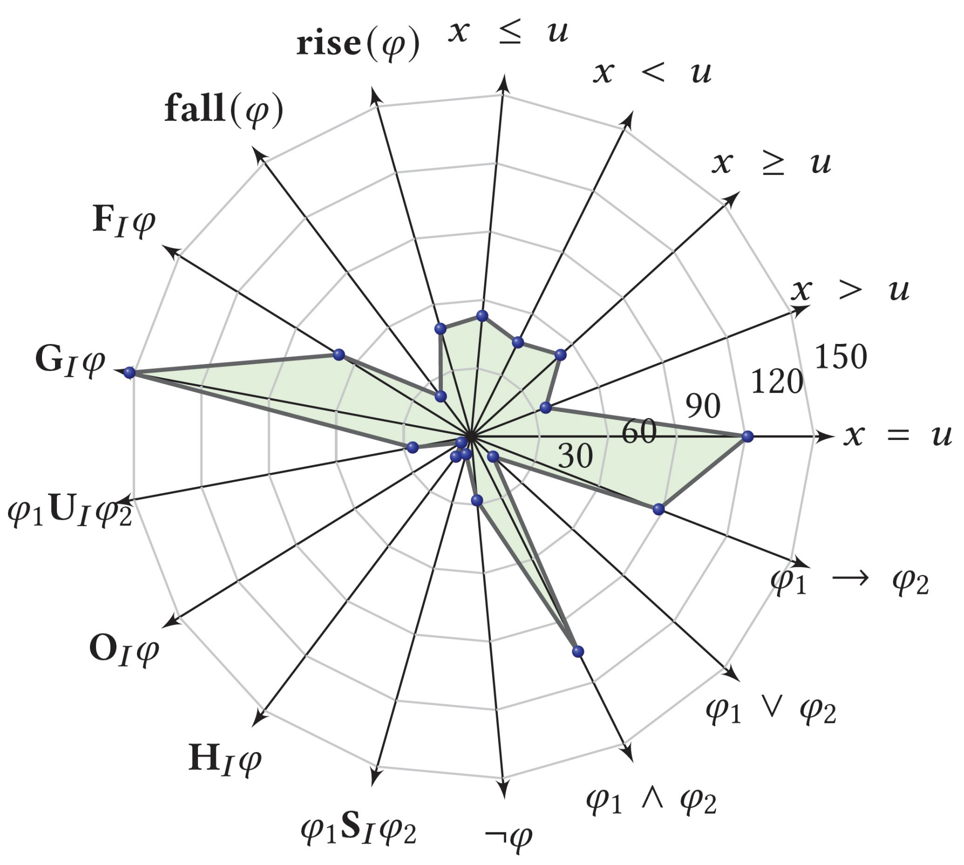
1. Restricted STL Fragment

- Not support complex grammar
- Not support ad-hoc templates in *other* group

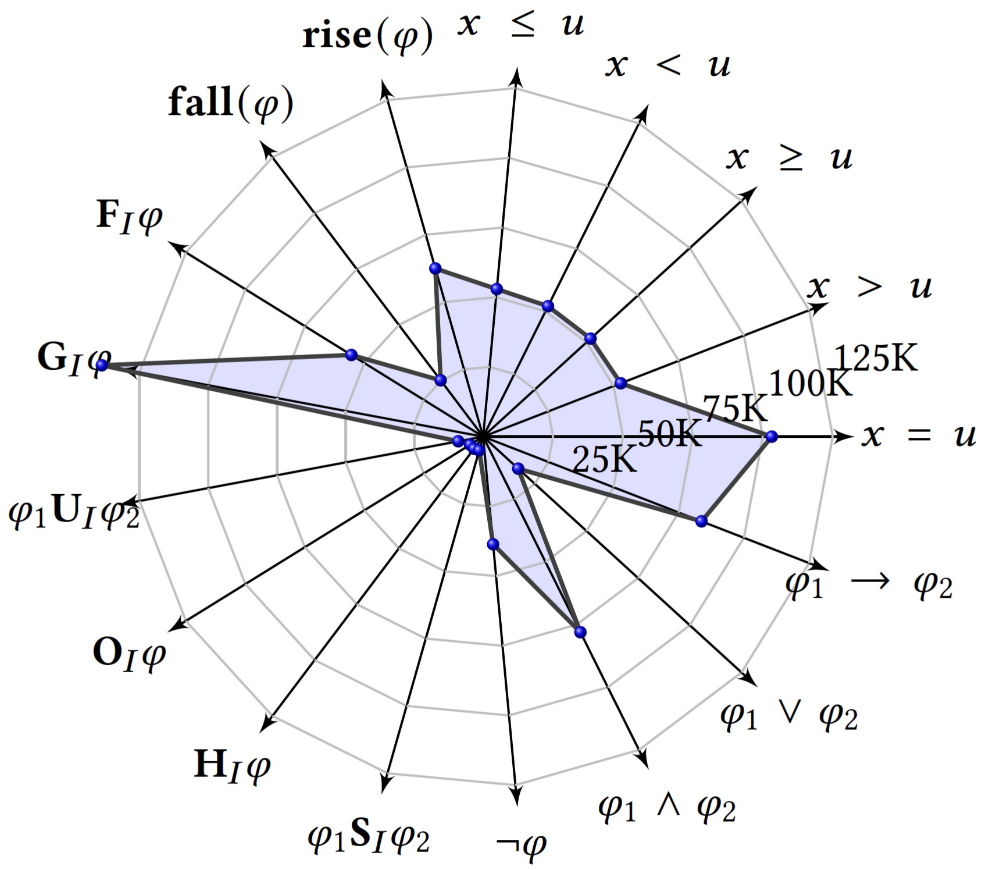


Corpus Construction – Generate STL Formulas

2. Compare to empirical analysis



Empirical statistics

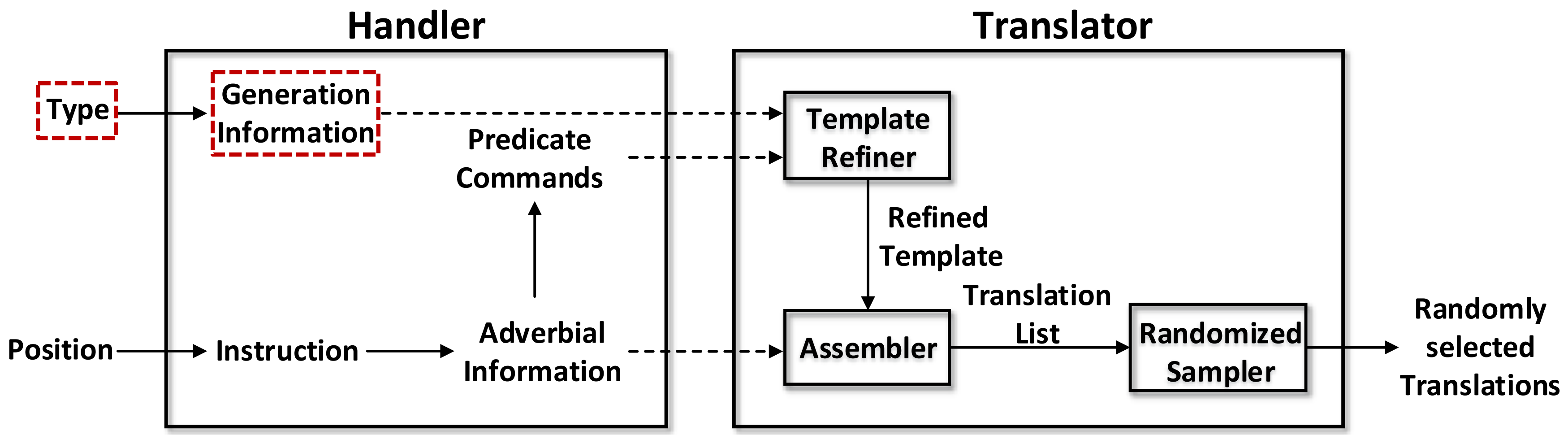


Generated Corpus

Corpus Construction – Translate STL into English

1. **Translate atomic propositions**

Example: $\mathbf{G}\,(\mathbf{rise}\,(\mathrm{In} > 0) \rightarrow \mathbf{rise}\,(\mathrm{Out} > 0))$



Type: event

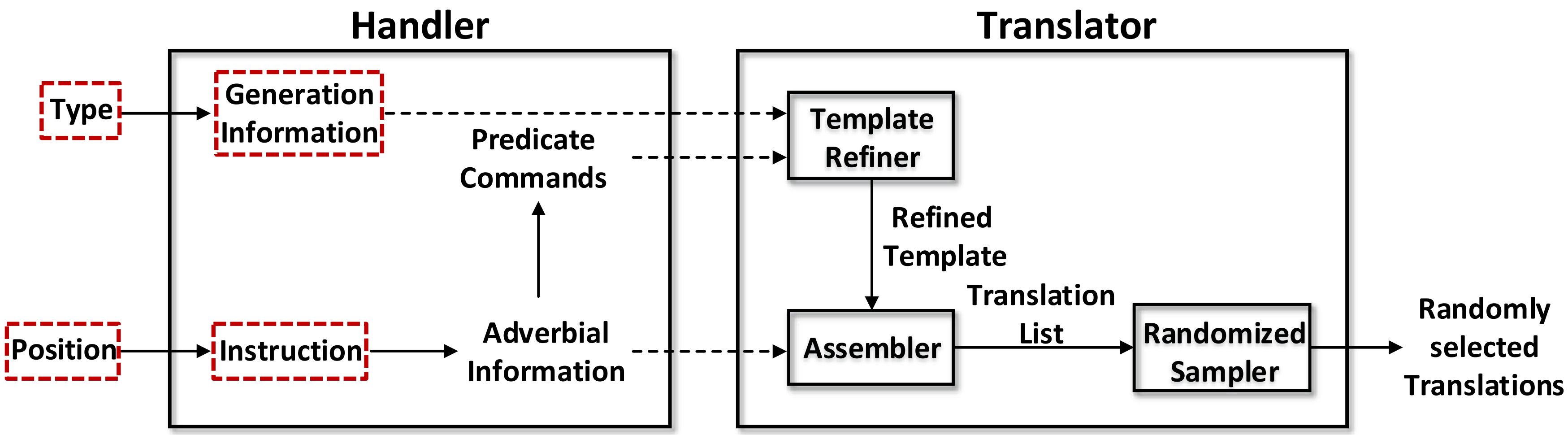
Generation information:

```
{'type': 'event',
 'index': [1, 1],
 'ingredient': ['Out', '0'],
 'expression': 'rise (Out > 0)'}
```

Corpus Construction – Translate STL into English

1. **Translate atomic propositions**

Example: $\mathbf{G}\ (\mathbf{rise}\ (\mathrm{In} > 0) \rightarrow \mathbf{rise}\ (\mathrm{Out} > 0))$



Position: `after_imply` →

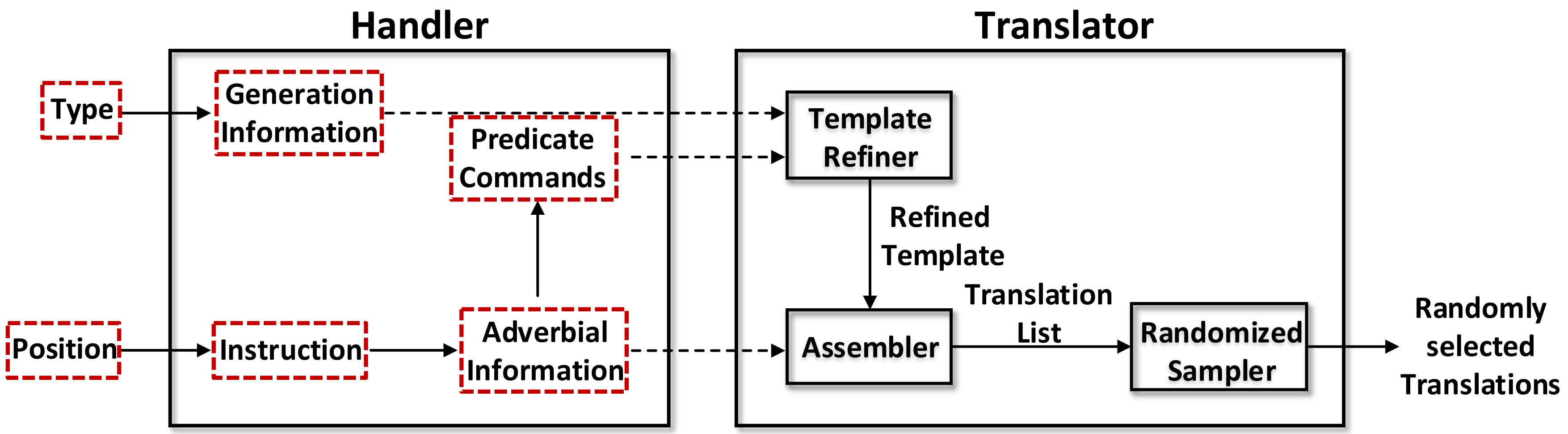
Instruction:

```
{'position': 'after_imply',
 'adverbial_query': 'adverbialEnabled'}
```

Corpus Construction – Translate STL into English

1. **Translate atomic propositions**

Example: $\mathbf{G}\,(\mathbf{rise}\,(\mathrm{In} > 0) \rightarrow \mathbf{rise}\,(\mathrm{Out} > 0))$



Adverbial Information:

```
{'adverb': ['immediately'],
 'adv_phrase': ['at once']}
```

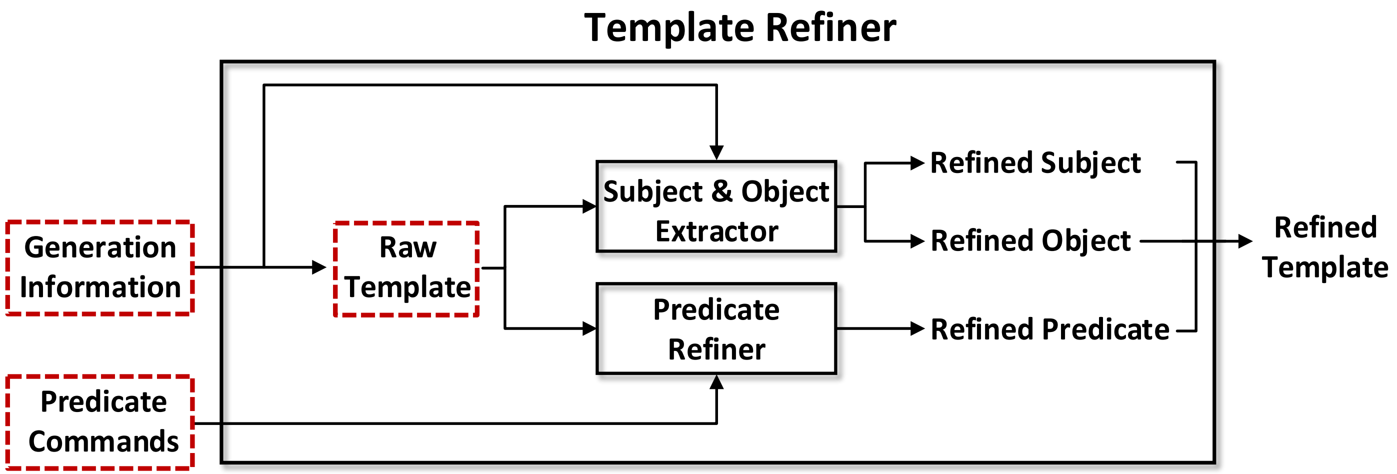
Predicate Commands:

```
{'simple_future_tense': ['will'],
 'modal_verb': ['should'],
 'adverb': ['immediately']}
```

Corpus Construction – Translate STL into English

1. Translate atomic propositions

Example: $\mathbf{G}\,(\mathbf{rise}\,(\mathrm{In} > 0) \rightarrow \mathbf{rise}\,(\mathrm{Out} > 0))$



Generation information:

```
{'type': 'event',
 'index': [1, 1],
 'ingredient': ['Out', '0'],
 'expression': 'rise (Out > 0)'}
```

Raw Template:

```
{'subject': ['sig', 'the value of sig'],
 'predicate': ['increase above',
               'become larger than'],
 'object': ['value']}
```

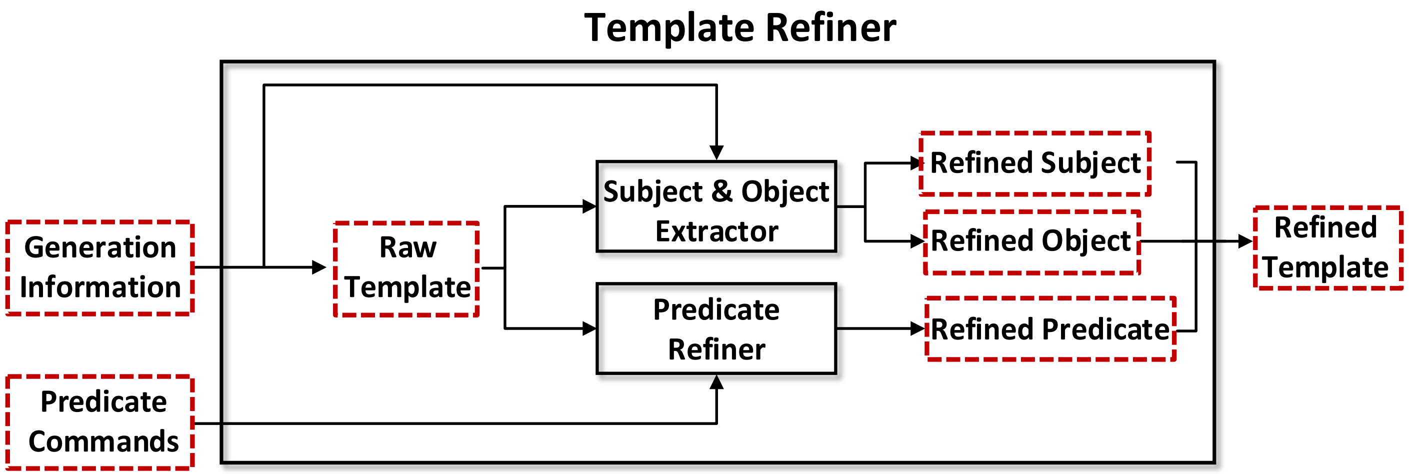
Predicate Commands:

```
{'simple_future_tense': ['will'],
 'modal_verb': ['should'],
 'adverb': ['immediately']}
```

Corpus Construction – Translate STL into English

1. Translate atomic propositions

Example: $\mathbf{G}\ (\mathbf{rise}\ (\mathrm{In} > 0) \rightarrow \mathbf{rise}\ (\mathrm{Out} > 0))$



Refined Subject:

- Out
- the value of Out

Refined Predicate:

- will increase above
- will immediately increase above
- should increase above
- should immediately increase above
- will become larger than
- will immediately become larger than
- should become larger than
- should immediately become larger than

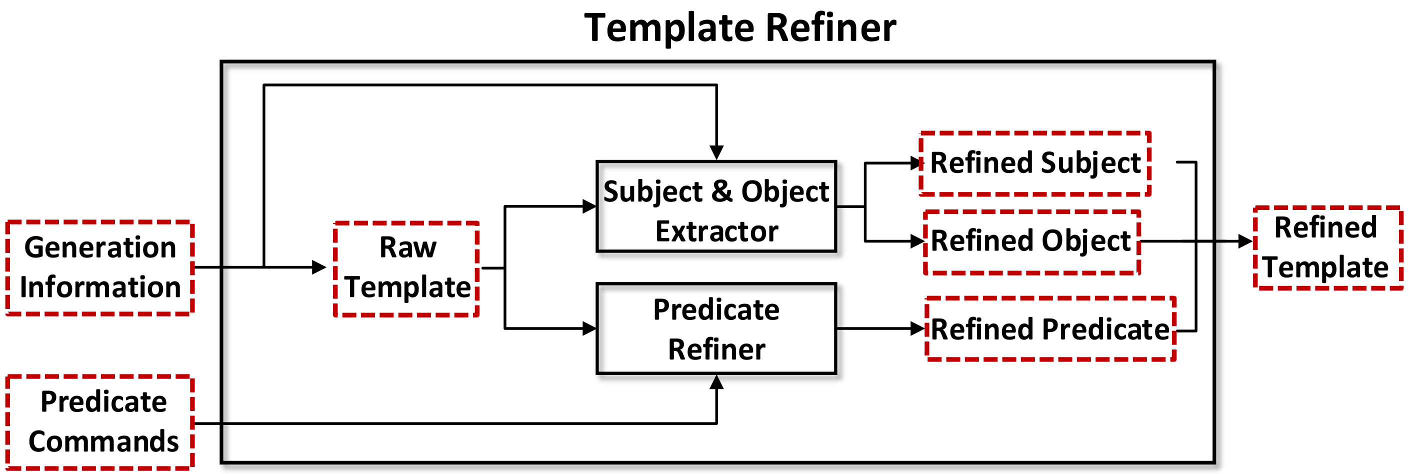
Refined Object:

- 0

Corpus Construction – Translate STL into English

1. **Translate atomic propositions**

Example: $\mathbf{G}\ (\mathbf{rise}\ (\mathrm{In} > 0) \rightarrow \mathbf{rise}\ (\mathrm{Out} > 0))$



Refined Subject:

- `Out`
- `the value of Out`

Refined Predicate:

- `will increase above`
- `will immediately increase above`
- `should increase above`
- `should immediately increase above`
- `will become larger than`
- `will immediately become larger than`
- `should become larger than`
- `should immediately become larger than`

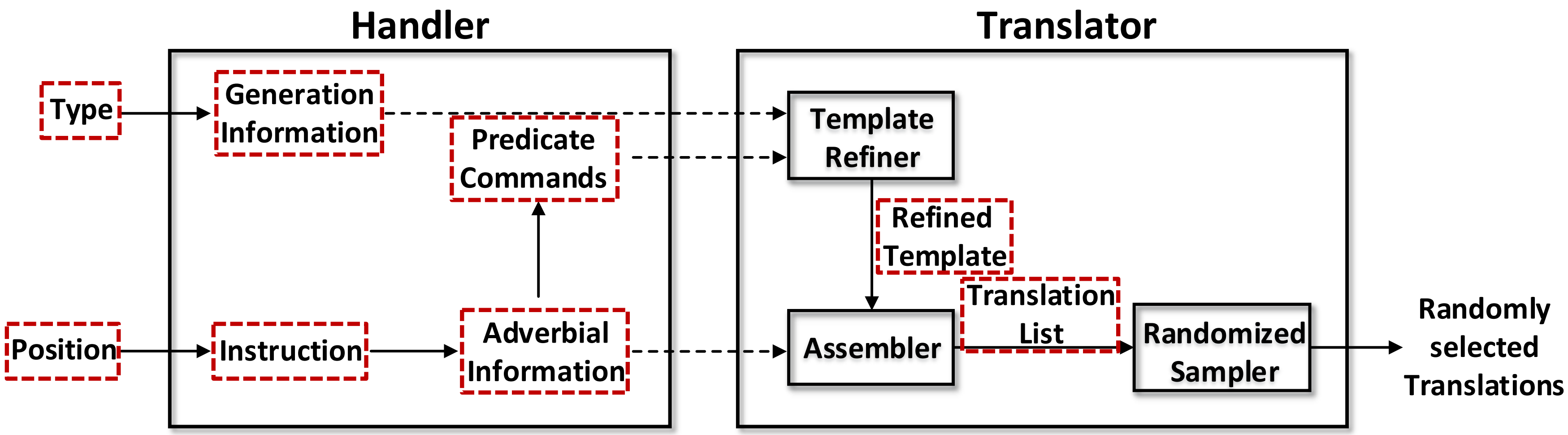
Refined Object:

- `0`

Corpus Construction – Translate STL into English

1. Translate atomic propositions

Example: $\mathbf{G}\,(\mathbf{rise}\,(\mathrm{In} > 0) \rightarrow \mathbf{rise}\,(\mathrm{Out} > 0))$



- **Adverbial Information:**

```
{'adverb': ['immediately'],
 'adv_phrase': ['at once']}
```

- **Refined Template:**

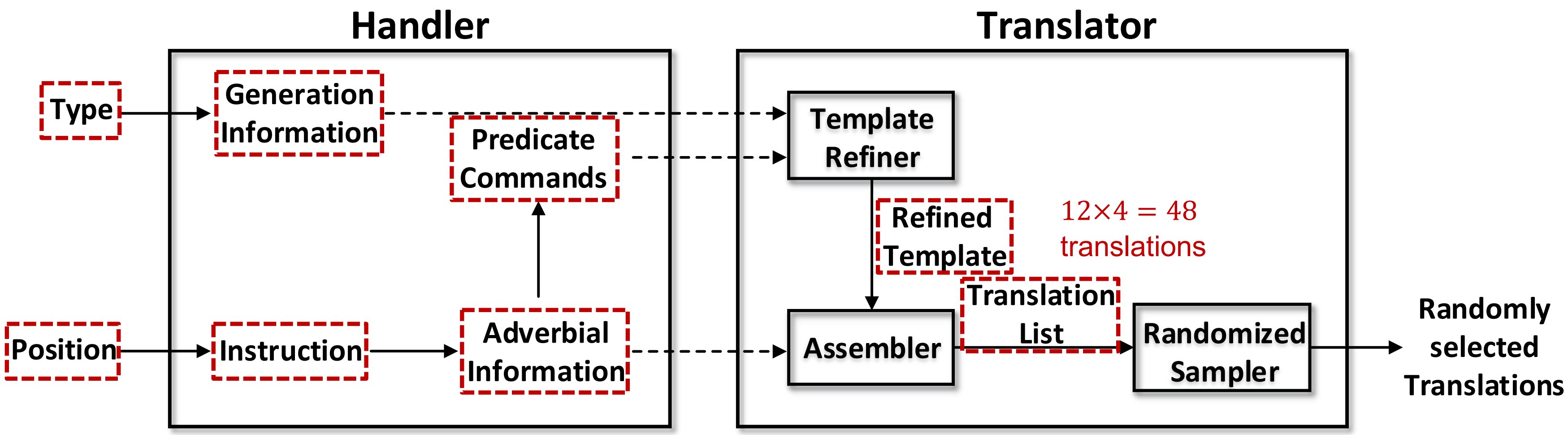
Assembler

```
1. the value of Out will increase above 0
2. the value of Out will immediately increase above 0
3. immediately the value of Out will increase above 0
4. the value of Out will increase above 0 immediately
5. at once the value of Out will increase above 0
6. the value of Out will increase above 0 at once
```

Corpus Construction – Translate STL into English

1. Translate atomic propositions

Example: $\mathbf{G}\,(\mathbf{rise}\,(\mathrm{In} > 0) \rightarrow \mathbf{rise}\,(\mathrm{Out} > 0))$



- **Adverbial Information:**

{'adverb': ['immediately'],
'adv_phrase': ['at once']}

- **Refined Template:**

Assembler

7. the value of Out should increase above 0
8. the value of Out should immediately increase above 0
9. immediately the value of Out should increase above 0
10. the value of Out should increase above 0 immediately
11. at once the value of Out should increase above 0
12. the value of Out should increase above 0 at once

Corpus Construction – Translate STL into English

2. Translate temporal phrases

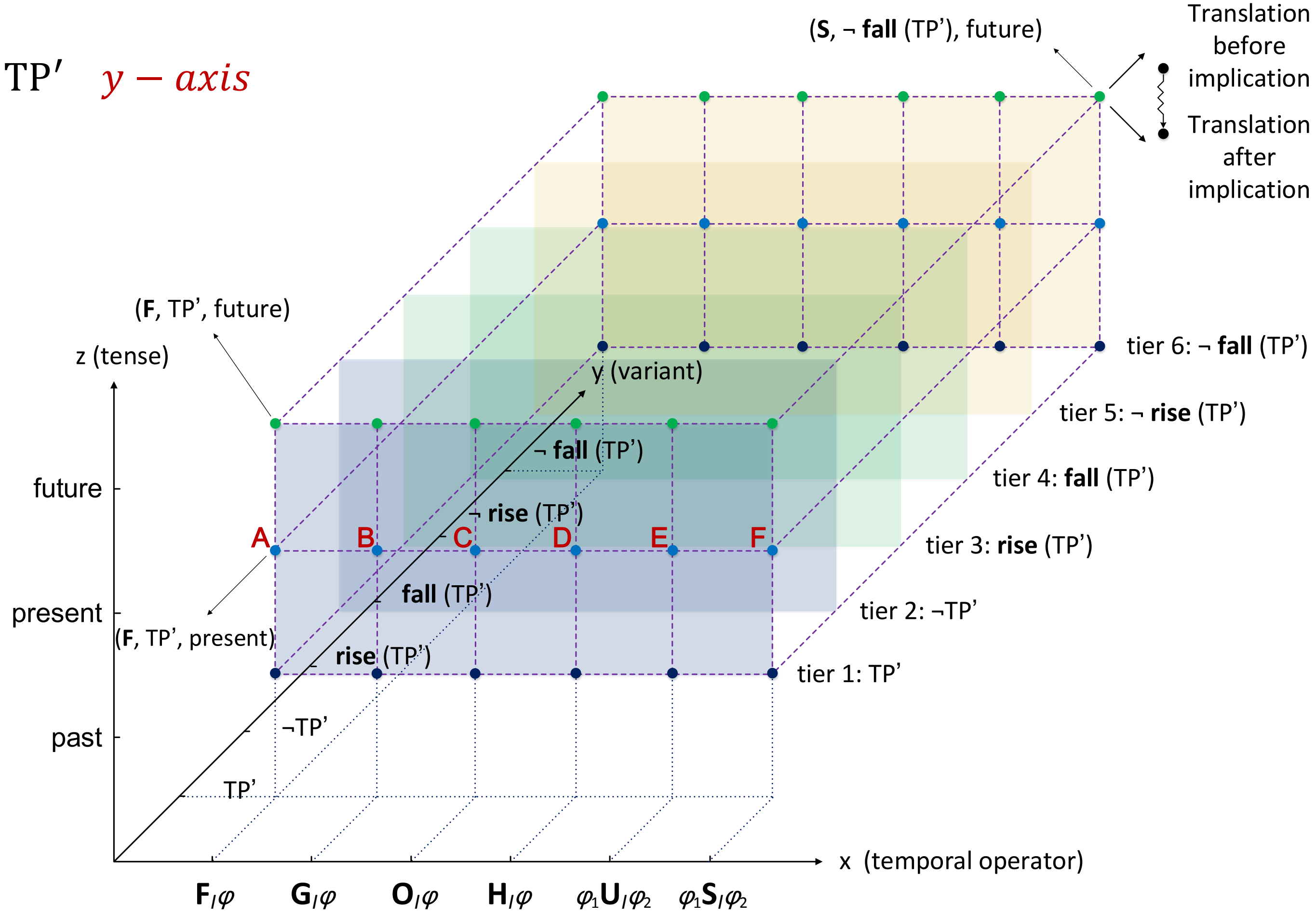
$$\mathrm{TP} \coloneqq \mathrm{TP}' \mid \neg\mathrm{TP}' \mid \textbf{rise}\ \mathrm{TP}' \mid \textbf{fall}\ \mathrm{TP}' \mid \neg\textbf{rise}\ \mathrm{TP}' \mid \neg\textbf{fall}\ \mathrm{TP}' \quad y-axis$$

$$\mathrm{TP}' \coloneqq \mathbf{UTO}_I(\alpha) \mid (\alpha)\mathbf{BTO}_I(\alpha) \quad x-axis$$

$$\mathbf{UTO} \in \{\mathbf{F}, \mathbf{G}, \mathbf{O}, \mathbf{H}\},\ \mathbf{BTO} \in \{\mathbf{U}, \mathbf{S}\},\ I \in \{t_1, t_2\}$$

General Strategy:

- Reuse translations of atomic propositions
- Add temporal adverbial modifiers
- Enrich verb tenses

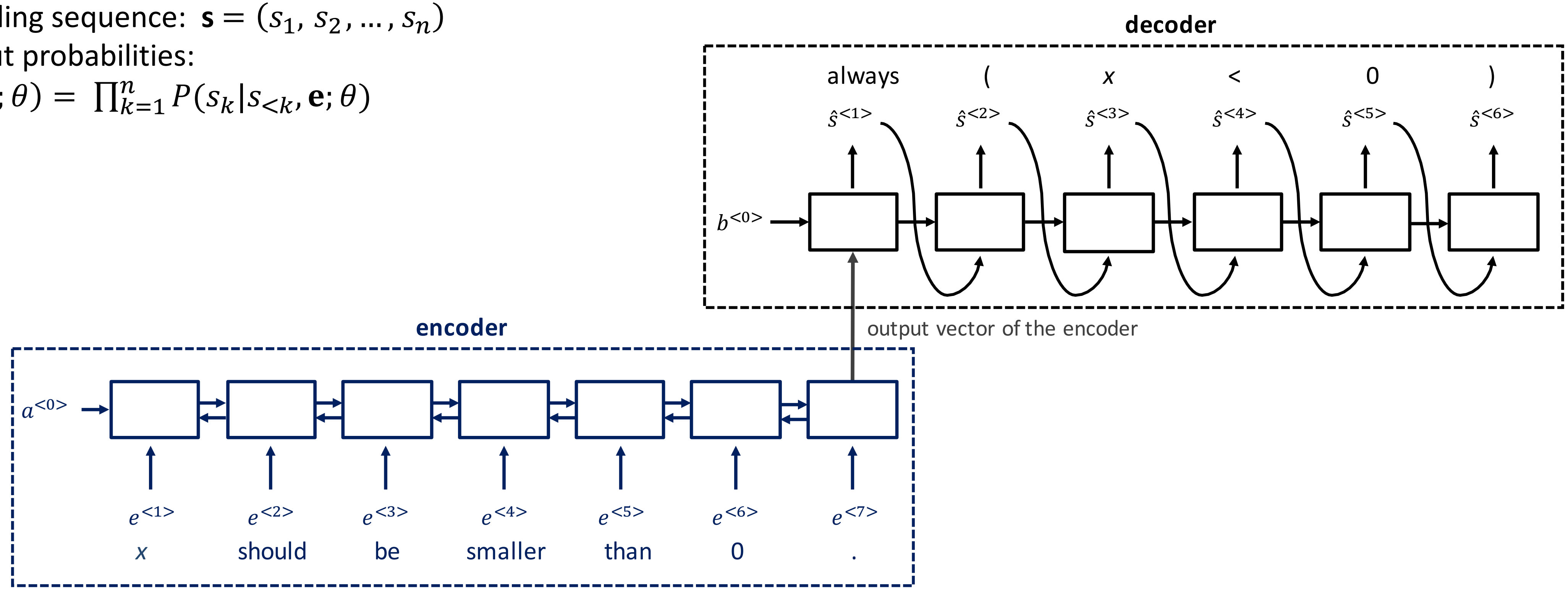


Step 3: Machine Translation

Machine Translation – AI Models

1. Seq2seq

- Encoding sequence: $\mathbf{e} = (e_1, e_2, \dots, e_m)$
- Decoding sequence: $\mathbf{s} = (s_1, s_2, \dots, s_n)$
- Output probabilities:
  $P(\mathbf{s}|\mathbf{e};\theta) = \prod_{k=1}^{n} P(s_k|s_{<k}, \mathbf{e};\theta)$

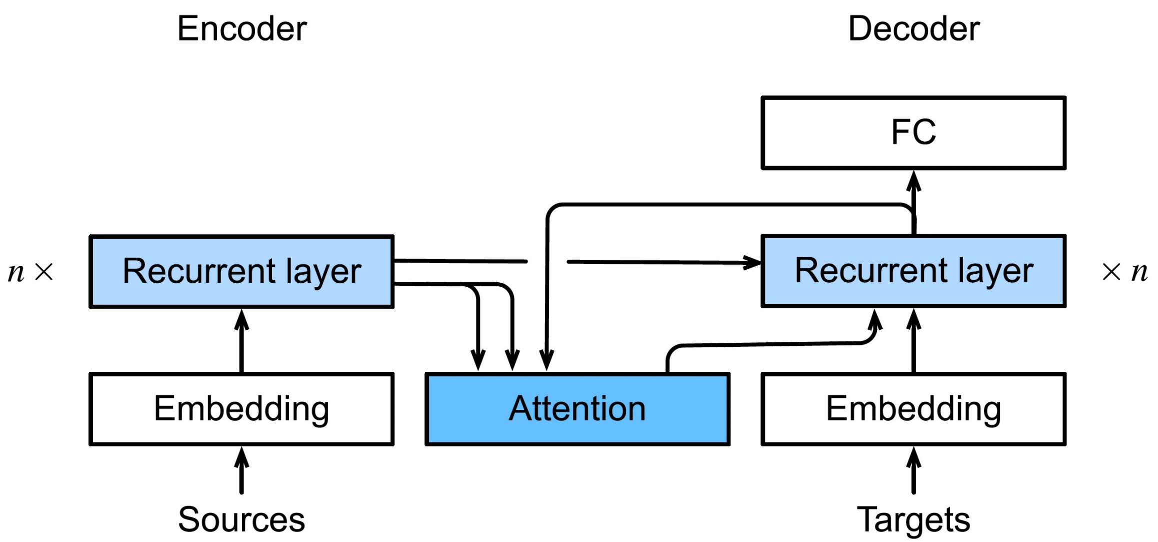


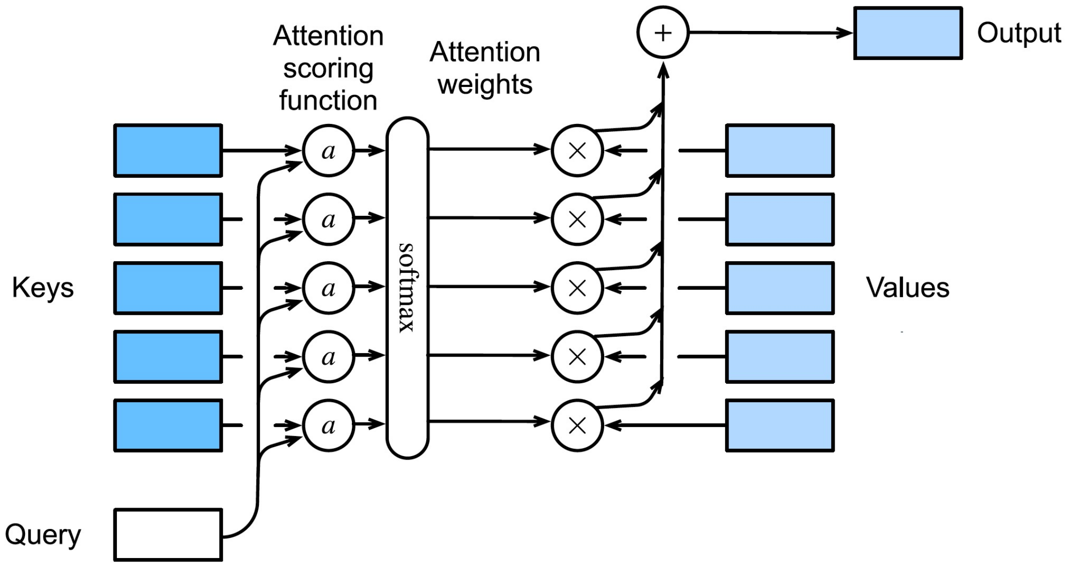
Encoder-decoder model using Recurrent neural network (RNN)

Machine Translation – AI Models

2. Att-seq2seq

- Encoding sequence: $\mathbf{e} = (e_1, e_2, \dots, e_m)$
- Decoding sequence: $\mathbf{s} = (s_1, s_2, \dots, s_n)$
- Output probabilities:
  $P(\mathbf{s}|\mathbf{e};\theta) = \prod_{k=1}^{n} P(s_k|s_{<k}, \mathbf{e};\theta)$





Bahdanau attention model[1]

[1] Zhang, Aston, et al. "Dive into deep learning." arXiv preprint arXiv:2106.11342 (2021).

Machine Translation – AI Models

3. Transformer (DeepSTL)

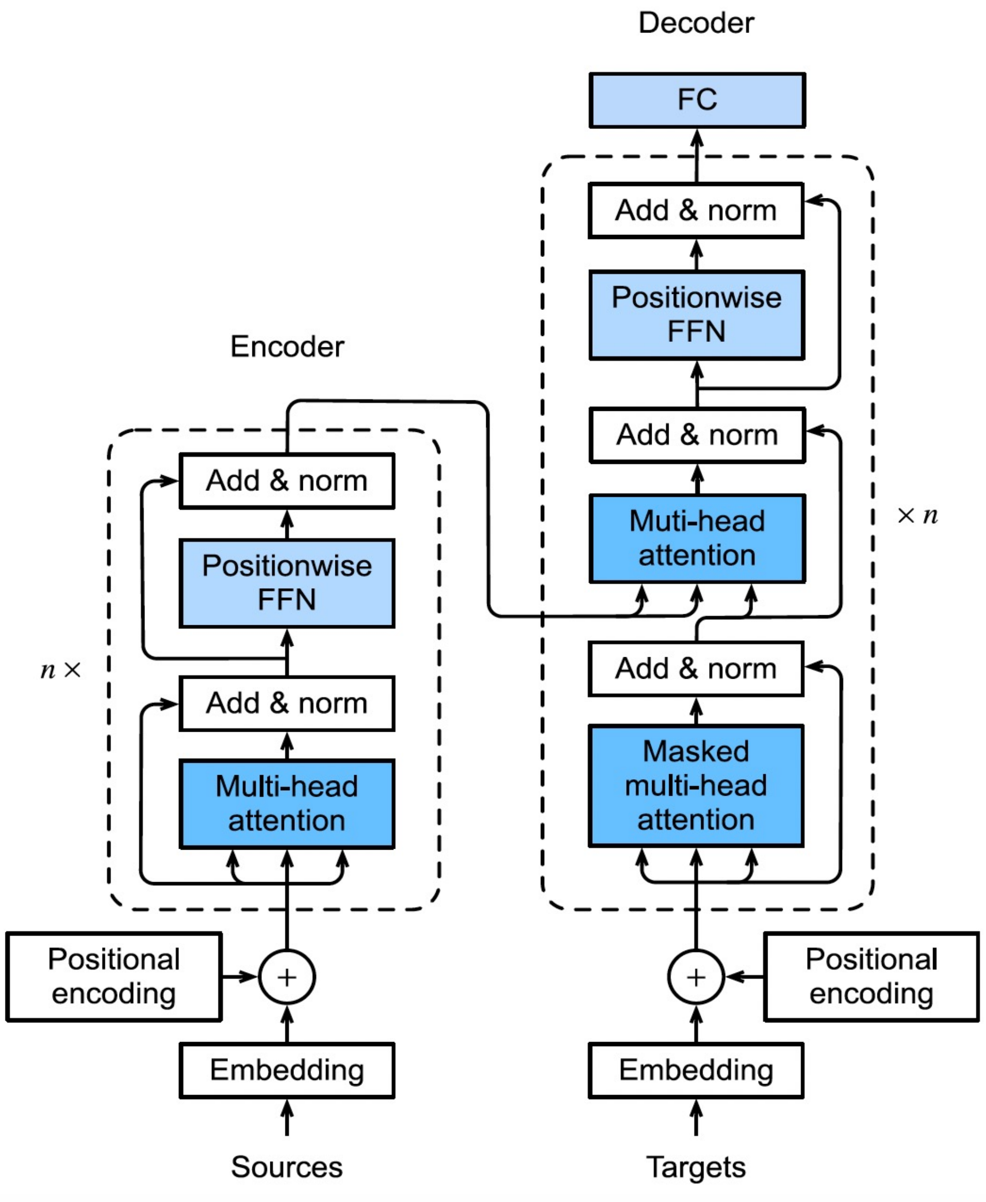
- Encoding sequence: $\mathbf{e} = (e_1, e_2, \dots, e_m)$
- Decoding sequence: $\mathbf{s} = (s_1, s_2, \dots, s_n)$
- Output probabilities:
  $P(\mathbf{s}|\mathbf{e};\theta) = \prod_{k=1}^{n} P(s_k|s_{<k}, \mathbf{e};\theta)$

Transformer's new features:

- Self-attention
- Multi-head attention
- Enable parallelization

Data split:

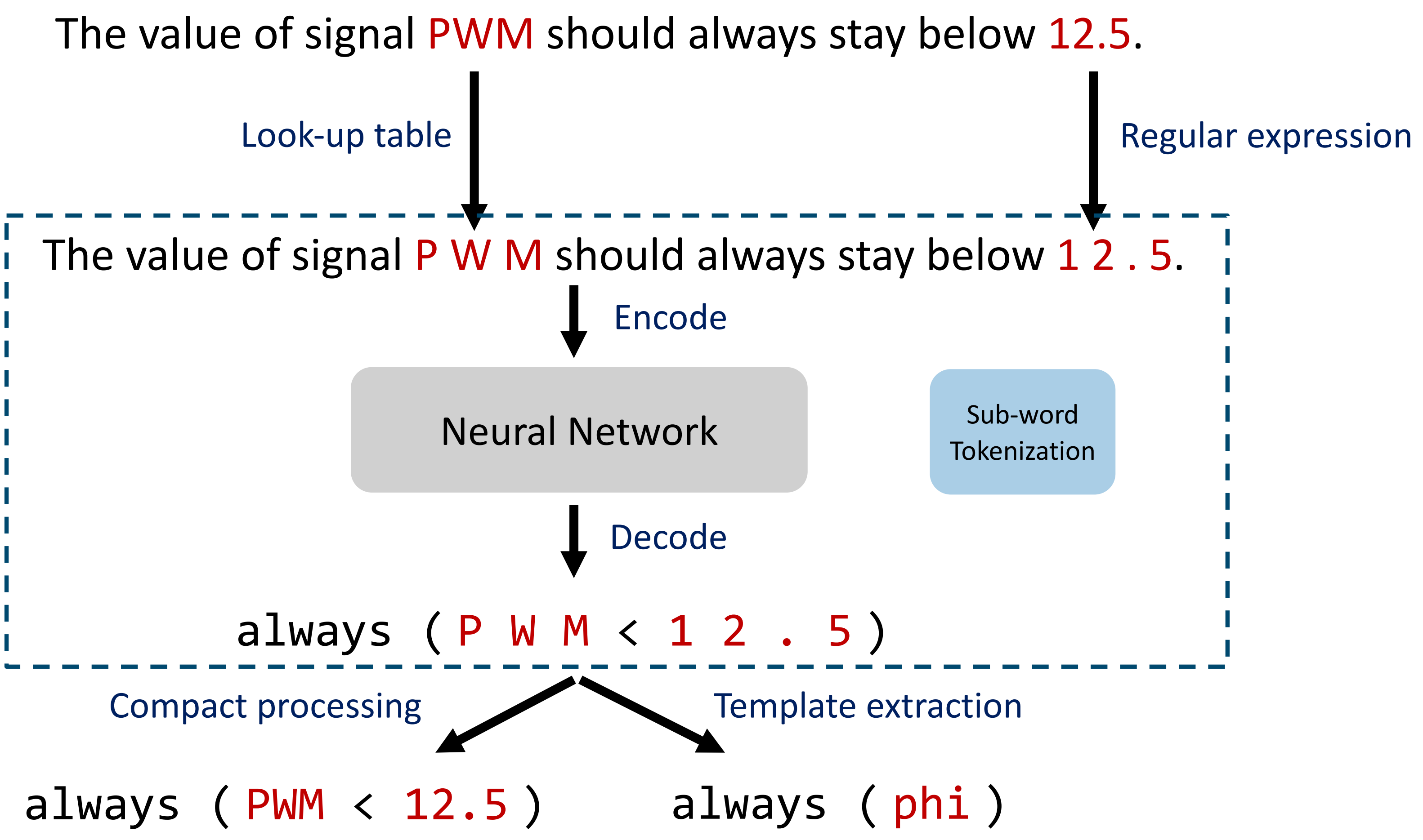
- Training set: 97,200 English-STL pairs
- Validation set: 10,800 English-STL pairs
- Testing set: 12,000 English-STL pairs



Transformer model[1]

Machine Translation – Process Identifiers and Constants

1. Pipeline



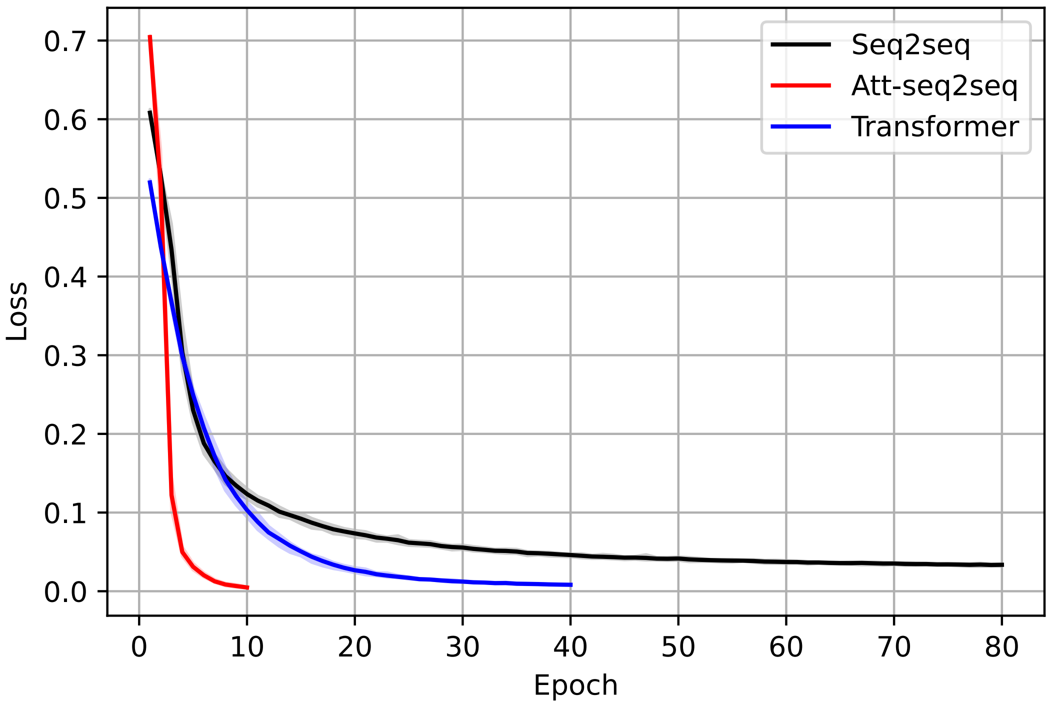
Accuracy Example:

Reference sequence:
always (P W M < 1 2 . 5)
Output sequence:
always (P W M < 1 2 . 6)
- **Formula accuracy: 10/11**

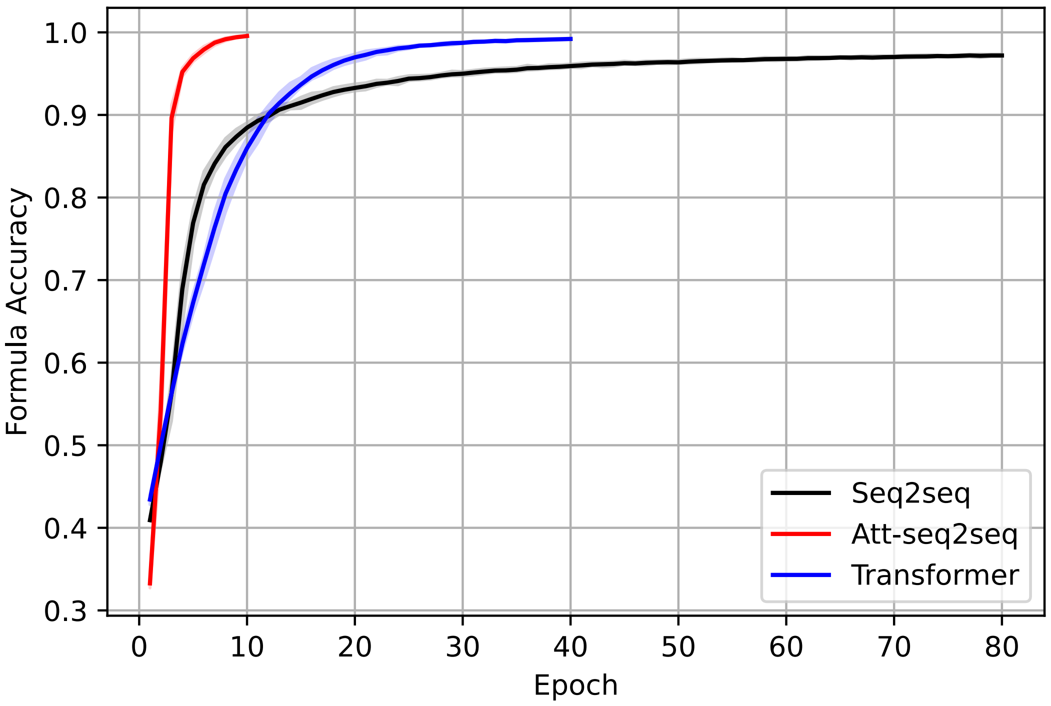
Reference template:
always (phi)
Output template:
always (phi)
- **Template accuracy: 1**

Machine Translation – Experimental Results

1. Training and Validation



Validation loss



Validation formula accuracy

Machine Translation – Experimental Results

2. Testing

Testing results on synthetic data

	Formula Acc.	Template Acc.	BLEU
Seq2seq	0.071 ± 0.0388	0.207 ± 0.0868	0.092 ± 0.0361
Att-seq2seq	0.977 ± 0.0060	0.980 ± 0.0063	0.996 ± 0.0011
Transformer	0.987 ± 0.0028	0.995 ± 0.0014	0.998 ± 0.0005

Testing results on 14 extrapolation cases

	Formula Acc.	Template Acc.	BLEU
Seq2seq	0.050 ± 0.0283	0.158 ± 0.0895	0.027 ± 0.0120
Att-seq2seq	0.559 ± 0.0865	0.742 ± 0.0660	0.888 ± 0.0348
Transformer	0.712 ± 0.0678	0.899 ± 0.0100	0.962 ± 0.0030

Machine Translation – Experimental Results

3. Translation results on extrapolation (Real output)

Example 1:

If the value of signal RWs_angular_momentum is greater than 0.35, then the value of signal RWs_torque shall be equal to 0.

- Transformer (C_t = −0.01393):
  `always ( RWs_angular_momentum > 0.35 -> RWs_torque == 0 )`

- Att-seq2seq (C_a = −0.30038):
  `always ( RWs_angular_mxyomemeEqm < 0.3 -> RWs_torque == 0 )`

- Seq2seq (C_s = −2.77145):
  `always ( WNcAi1iDSDDyD1yD2y171a71aa2345324621 ) 5` ......
  too long, display omitted

Machine Translation – Experimental Results

3. Translation results on extrapolation (Real output)

Example 2:

Whenever Op_Cmd changes to Passive then in response Spd_Act changes to 0 after at most 500 time units.

- Transformer (C_t = −0.00091):
  `always ( rise ( Op_Cmd == Passive ) -> eventually [ 0 : 500 ] ( rise ( Spd_Act == 0 ) ) )`

- Att-seq2seq (C_a = −0.10360):
  `always ( rise ( Op_Cmd == Passive ) -> not ( eventually [ 0 : 500 ] ( Spd_Act == 0 ) ) )`

- Seq2seq (C_s = −3.03260):
  `always ( rise (PIweD > 12.3 Q8y5yDy6y1y1R11y1y1g1y1A` ......
  too long, display omitted

Machine Translation – Experimental Results

3. Translation results on extrapolation (Real output)

Example 3:

Whenever V_Mot enters the range [1, 12] then in response starting after at most 100 time units Spd_Act must be in the range [100, 1000].

- Transformer (C_t = −0.00873):
  `always ( rise ( V_Mot >= 1 and V_Mot <= 12 ) -> eventually [ 0 : 100 ] ( Spd_Act >= 100 and Spd_Act <= 1000 ) )`

- Att-seq2seq (C_a = −0.06080):
  `always ( rise ( V_Mot >= 1 and V_Mot <= 12 ) -> not ( eventually [ 0 : 100 ] ( Spd_Act >= 100 and Spd_Act <= 1000 ) ) )`

- Seq2seq (C_s = −2.68981):
  `always ( rise ( p_qHX > 4 Q3DaQaDamymaOlQ ) ya ) 4 fall` ......
  too long, display omitted

Future Work

Future Work

1. Data augmentation
2. Introduce domain knowledge
3. Name entity recognition (NER)
4. Improve training procedure
5. Explore attention mechanism
6. Neural symbolic approach
7. Augment interaction

Thank you