Logical Story Representations via FrameNet + Semantic Parsing

Lane Lawley and Lenhart Schubert

Schema/Script Representations

FrameNet:

- Rich semantics (inheritance, composition)
- Comprehensive semantic roles
- Difficult to create new frames
- Not directly usable for inference

Locating

Definition:

A Perceiver determines the Location of a Sought_entity within a Ground.
Kim FOUND his hat on the far side.

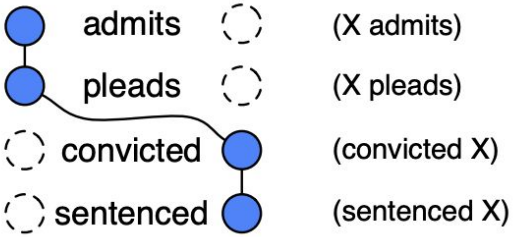
FEs:

Core:

Ground []	The general set of entities that the Perceiver examined in order to find the Sought_entity. Sam LOCATED the Imperial regal among the coins in his pocket and passed it to the boy.
Location [Loc] **Semantic Type:** Location	The position of the Sought_entity. I FOUND him under the bed.
Perceiver [] **Semantic Type:** Sentient	The individual that determines the Location of the Sought_entity. I finally FOUND the peanut underneath the couch cusion.
Sought_entity [] **Semantic Type:** Physical_entity **Non-Core:**	The entity that the Perceiver succeeds in finding.
Time [tim]	The time when the Sought_entity is located. We FOUND it last night.

Chambers & Jurafsky-style scripts:

- Easy to acquire from large text corpora
- Inexpressive event representation
- Simple, linear structure



Episodic Logic (EL) schemas

EL schemas use a rich, formal logic designed to closely resemble English syntax.

```
Gaurav looked for Mariel.
```

```
(GAURAV ((ADV-A (FOR.P MARIEL)) LOOK.V))
```

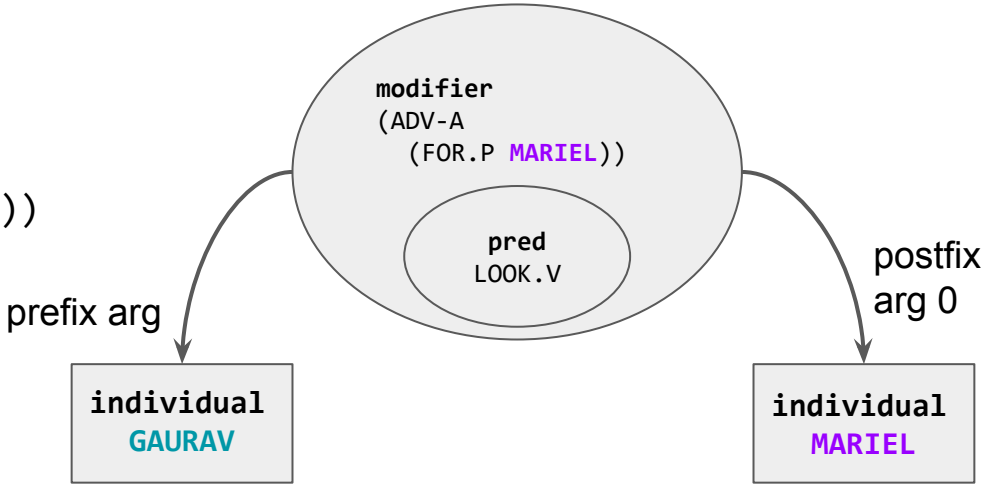
Schemas are collections of expressive logical formulas with shared variables to aid with inference.

Episodic Logic (EL) schemas

EL schemas use a rich, formal logic designed to closely resemble English syntax.

```
Gaurav looked for Mariel.

(GAURAV ((ADV-A (FOR.P MARIEL)) LOOK.V))
```



Schemas are collections of expressive logical formulas with shared variables to aid with inference.

```
(EPI-SCHEMA ((?X GO.2.V ?L) ** ?E)
      :Roles
            !R1 (?B BOOK.N)
            !R2 (?X PERSON.N)
            !R3 (?L LIBRARY.N)

      :Steps
            ?E1 (?X GO.V ?L)
            ?E2 (?X ((ADV-A (FOR.P ?B)) LOOK.V))
            ?E3 (?X FIND.V ?B)
            ?E4 (?X READ.V ?B)

      :Episode-relations
            !W1 (?E1 BEFORE ?E2)
            !W2 (?E2 BEFORE ?E3)
            !W3 (?E3 BEFORE ?E4)

      :Subordinate-constraints
            !S1 ((?E3-> ?Y) = ?L)
)
```

```
(EPI-SCHEMA ((?X GO.2.V ?L) ** ?E)
      :Roles
            !R1 (?B BOOK.N)
            !R2 (?X PERSON.N)
            !R3 (?L LIBRARY.N)

      :Steps
            ?E1 (?X GO.V ?L)
            ?E2 (?X ((ADV-A (FOR.P ?B)) LOOK.V))
            ?E3 (?X FIND.V ?B)
            ?E4 (?X READ.V ?B)

      :Episode-relations
            !W1 (?E1 BEFORE ?E2)
            !W2 (?E2 BEFORE ?E3)
            !W3 (?E3 BEFORE ?E4)

      :Subordinate-constraints
            !S1 ((?E3-> ?Y) = ?L)
)
```

```
(EPI-SCHEMA ((?X GO.2.V ?L) ** ?E)
      :Roles
            !R1 (?B BOOK.N)
            !R2 (?X PERSON.N)
            !R3 (?L LIBRARY.N)

      :Steps
            ?E1 (?X GO.V ?L)
            ?E2 (?X ((ADV-A (FOR.P ?B)) LOOK.V))
            ?E3 (?X FIND.V ?B)
            ?E4 (?X READ.V ?B)

      :Episode-relations
            !W1 (?E1 BEFORE ?E2)
            !W2 (?E2 BEFORE ?E3)
            !W3 (?E3 BEFORE ?E4)

      :Subordinate-constraints
            !S1 ((?E3-> ?Y) = ?L)
)
```

```
(EPI-SCHEMA ((?X GO.2.V ?L) ** ?E)
      :Roles
            !R1 (?B BOOK.N)
            !R2 (?X PERSON.N)
            !R3 (?L LIBRARY.N)

      :Steps
            ?E1 (?X GO.V ?L)
            ?E2 (?X ((ADV-A (FOR.P ?B)) LOOK.V))
            ?E3 (?X FIND.V ?B)
            ?E4 (?X READ.V ?B)

      :Episode-relations
            !W1 (?E1 BEFORE ?E2)
            !W2 (?E2 BEFORE ?E3)
            !W3 (?E3 BEFORE ?E4)

      :Subordinate-constraints
            !S1 ((?E3-> ?Y) = ?L)
)
```

```
(EPI-SCHEMA ((?P FIND.V ?O) ** ?E)
      :Roles
            !R2 (?P PERSON.N)
            !R3 (?O OBJECT.N)
            !R3 (?Y LOCATION.N)

      :Preconditions
            ?I1 (?O AT.P ?Y)
            ?I1 (?P AT.P ?Y)
            ?P1 (NOT (?P KNOW.V (THAT (?O AT.P ?Y))))

      :Postconditions
            ?P1 (?P KNOW.V (THAT (?O AT.P ?Y)))
)
```

```
(EPI-SCHEMA ((?X GO.2.V ?L) ** ?E)
      :Roles
            !R1 (?B BOOK.N)
            !R2 (?X PERSON.N)
            !R3 (?L LIBRARY.N)

      :Steps
            ?E1 (?X GO.V ?L)
            ?E2 (?X ((ADV-A (FOR.P ?B)) LOOK.V))
            ?E3 (?X FIND.V ?B)
            ?E4 (?X READ.V ?B)

      :Episode-relations
            !W1 (?E1 BEFORE ?E2)
            !W2 (?E2 BEFORE ?E3)
            !W3 (?E3 BEFORE ?E4)

      :Subordinate-constraints
            !S1 ((?E3-> ?Y) = ?L)
)
```

```
(EPI-SCHEMA ((?P FIND.V ?O) ** ?E)
      :Roles
            !R2 (?P PERSON.N)
            !R3 (?O OBJECT.N)
            !R3 (?Y LOCATION.N)

      :Preconditions
            ?I1 (?O AT.P ?Y)
            ?I1 (?P AT.P ?Y)
            ?P1 (NOT (?P KNOW.V (THAT (?O AT.P ?Y))))

      :Postconditions
            ?P1 (?P KNOW.V (THAT (?O AT.P ?Y)))
)
```

```
(EPI-SCHEMA ((?X GO.2.V ?L) ** ?E)
      :Roles
            !R1 (?B BOOK.N)
            !R2 (?X PERSON.N)
            !R3 (?L LIBRARY.N)

      :Steps
            ?E1 (?X GO.V ?L)
            ?E2 (?X ((ADV-A (FOR.P ?B)) LOOK.V))
            ?E3 (?X FIND.V ?B)
            ?E4 (?X READ.V ?B)

      :Episode-relations
            !W1 (?E1 BEFORE ?E2)
            !W2 (?E2 BEFORE ?E3)
            !W3 (?E3 BEFORE ?E4)

      :Subordinate-constraints
            !S1 ((?E3-> ?Y) = ?L)
)
```

Bootstrap schema learning with simple "protoschemas" (e.g. `FIND.V`)

```
(EPI-SCHEMA ((?P FIND.V ?O) ** ?E)
      :Roles
            !R2 (?P PERSON.N)
            !R3 (?O OBJECT.N)
            !R3 (?Y LOCATION.N)

      :Preconditions
            ?I1 (?O AT.P ?Y)
            ?I1 (?P AT.P ?Y)
            ?P1 (NOT (?P KNOW.V (THAT (?O AT.P ?Y))))

      :Postconditions
            ?P1 (?P KNOW.V (THAT (?O AT.P ?Y)))
)
```

Protoschemas

- Biological functions (eating, sleeping, etc.)
- Self-motion
- Transportation of objects
- Possession
- Object manipulation
- Social interaction (questions, asking for favors, etc.)

Schema Learning

Bootstrapping with protoschemas lets us progressively build more complex schemas…

Schema Learning

Bootstrapping with protoschemas lets us progressively build more complex schemas…

…but how do we get the initial protoschemas from text?

Protoschema Invocation

```
(?P TRAVEL.V ?D)
```

Protoschema Invocation

```
(?P TRAVEL.V ?D)
```

He went home.

Protoschema Invocation

```
(?P TRAVEL.V ?D)
```

He went home.

He ran to the store.

Protoschema Invocation

```
(?P TRAVEL.V ?D)
```

He went home.

He ran to the store.

He was on his way.

Protoschema Invocation

```
(?P TRAVEL.V ?D)
```

We need a non-brittle NL-to-schema invocation identifier.

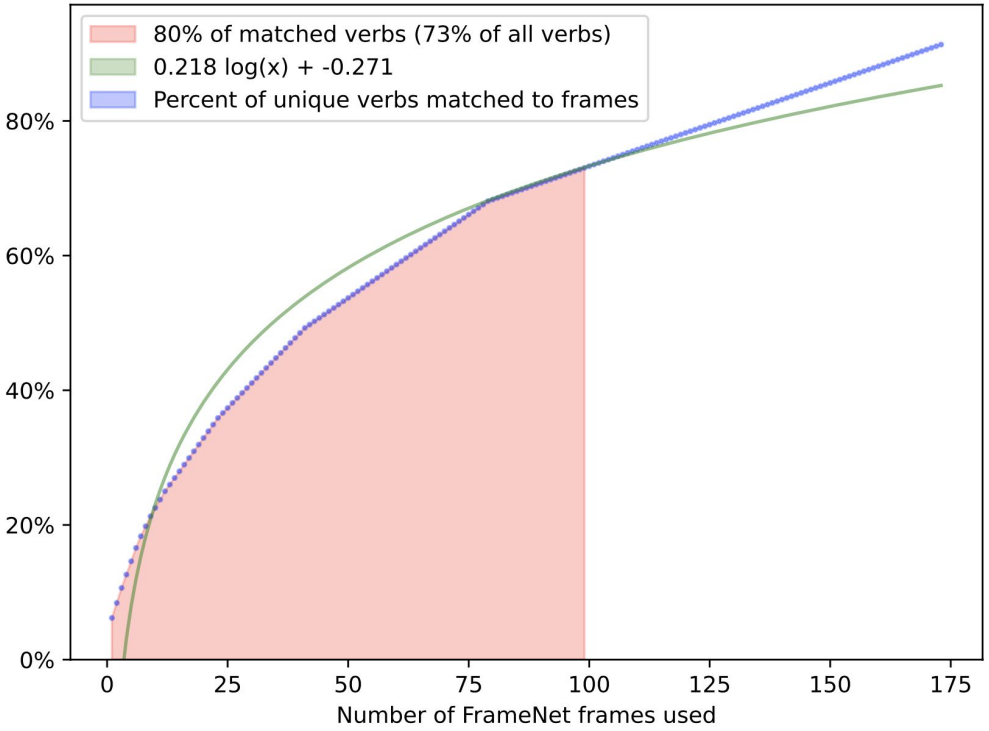
He went home.

He ran to the store.

He was on his way.

Matching Protoschemas w/ FrameNet

Like how protoschemas should cover most actions, a relatively small set of FN frames covers a relatively large number of verbs.



Matching Protoschemas w/ FrameNet

These most frequent FN frames inspire some basic protoschemas:

Motion

Self_motion

Experiencer_focus

Perception_experience

Arriving

Locating

Request

Bringing

Ingestion

...

Matching Protoschemas w/ FrameNet

These most frequent FN frames inspire some basic protoschemas:

Motion

Self_motion

Experiencer_focus

Perception_experience

Arriving

Locating

Request

Bringing

Ingestion

...

Matching Protoschemas w/ FrameNet

These most frequent FN frames inspire some basic protoschemas:

Locating

Definition:

A Perceiver determines the Location of a Sought_entity within a Ground.
Kim FOUND his hat on the far side.

FEs:

Core:

Ground []	The general set of entities that the Perceiver examined in order to find the Sought_entity. Sam LOCATED the Imperial regal among the coins in his pocket and passed it to the boy.
Location [Loc] **Semantic Type:** Location	The position of the Sought_entity. I FOUND him under the bed.
Perceiver [] **Semantic Type:** Sentient	The individual that determines the Location of the Sought_entity. I finally FOUND the peanut underneath the couch cusion.
Sought_entity [] **Semantic Type:** Physical_entity	The entity that the Perceiver succeeds in finding.
Non-Core:	
Time [tim]	The time when the Sought_entity is located. We FOUND it last night.

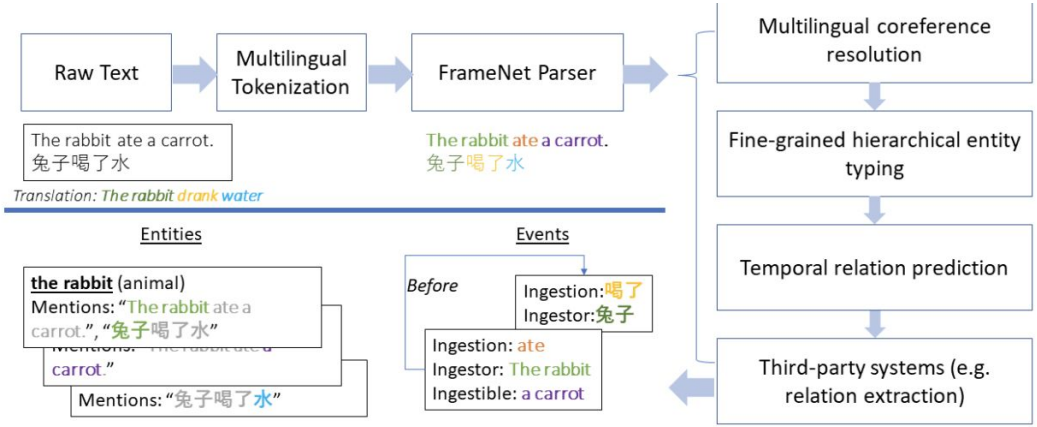
```
(EPI-SCHEMA ((?P FIND.V ?O) ** ?E)
      :Roles
            !R2 (?P PERSON.N)
            !R3 (?O OBJECT.N)
            !R3 (?Y LOCATION.N)

      :Preconditions
            ?I1 (?O AT.P ?Y)
            ?I1 (?P AT.P ?Y)
            ?P1 (NOT (?P KNOW.V (THAT (?O AT.P ?Y))))

      :Postconditions
            ?P1 (?P KNOW.V (THAT (?O AT.P ?Y)))
)
```

Using FrameNet parsing to match protoschemas

LOME is a state-of-the-art information extraction system with FrameNet parsing at its core.

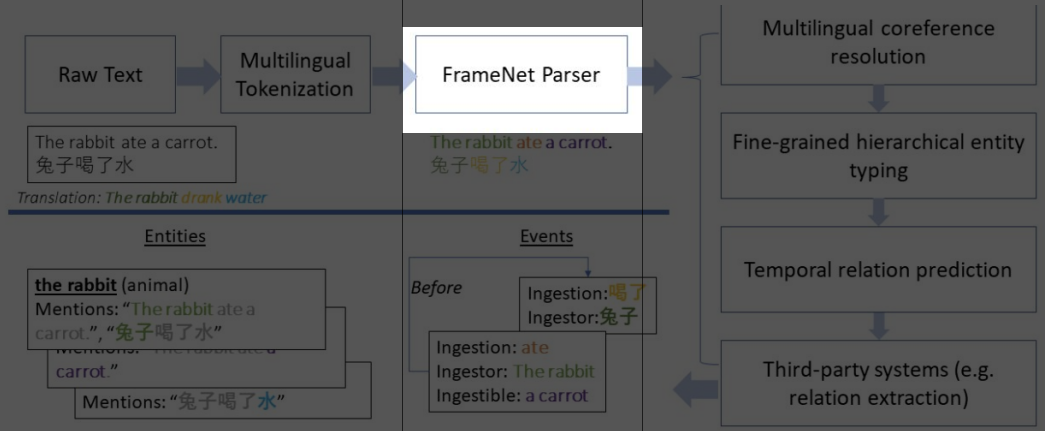


LOME: Large Ontology Multilingual Extraction

Patrick Xia[1]*, Guanghui Qin[1]*, Siddharth Vashishtha[2]
Yunmo Chen[1], Tongfei Chen[1], Chandler May[1], Craig Harman[1]
Kyle Rawlins[1], Aaron Steven White[2], Benjamin Van Durme[1]
[1] Johns Hopkins University, [2] University of Rochester
{paxia,qin,vandurme}@jhu.edu

Using FrameNet parsing to match protoschemas

LOME is a state-of-the-art information extraction system with FrameNet parsing at its core.



LOME: Large Ontology Multilingual Extraction

Patrick Xia[1]*, Guanghui Qin[1]*, Siddharth Vashishtha[2]
Yunmo Chen[1], Tongfei Chen[1], Chandler May[1], Craig Harman[1]
Kyle Rawlins[1], Aaron Steven White[2], Benjamin Van Durme[1]
[1] Johns Hopkins University, [2] University of Rochester
{paxia,qin,vandurme}@jhu.edu

This Work

1. Align FN parses with EL parses, situating semantic roles in logical domain

2. Map enriched FN frames to EL protoschemas

Aligning FN parses with EL parses

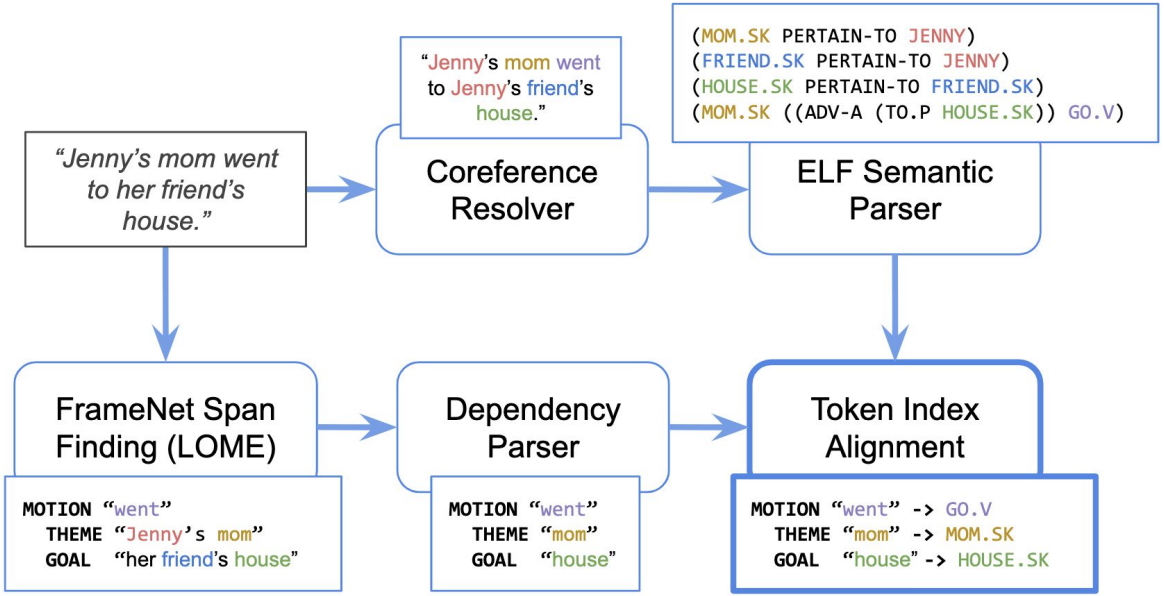


Figure 2: The architecture of the system. Raw story text is fed along two tracks: the logical-semantic parsing track, shown along the top, and the FrameNet parsing track, shown along the bottom. The FrameNet text spans are reduced to direct object tokens and correlated with logical individuals in the ELF parse via token index matching.

This Work: Aligning FN parses with EL parses

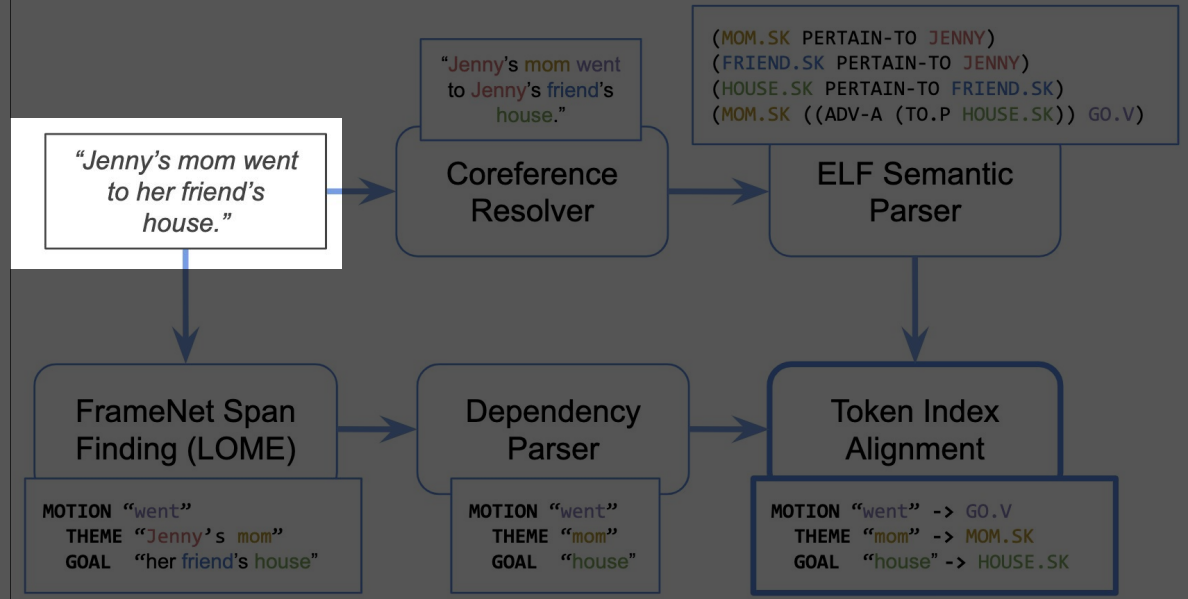


Figure 2: The architecture of the system. Raw story text is fed along two tracks: the logical-semantic parsing track, shown along the top, and the FrameNet parsing track, shown along the bottom. The FrameNet text spans are reduced to direct object tokens and correlated with logical individuals in the ELF parse via token index matching.

This Work: Aligning FN parses with EL parses

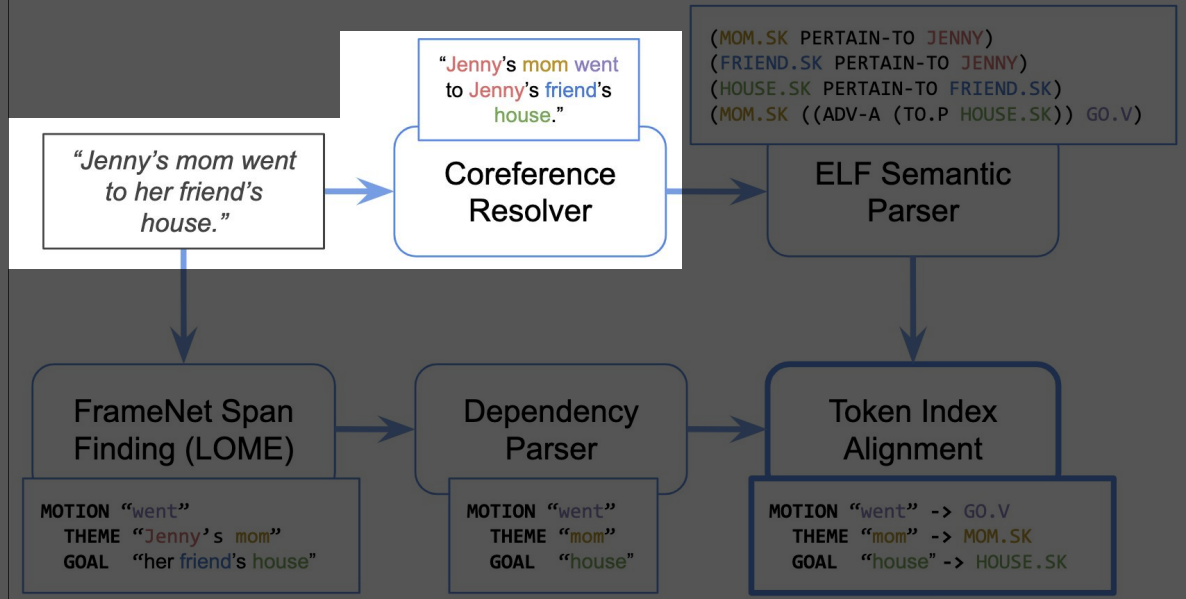


Figure 2: The architecture of the system. Raw story text is fed along two tracks: the logical-semantic parsing track, shown along the top, and the FrameNet parsing track, shown along the bottom. The FrameNet text spans are reduced to direct object tokens and correlated with logical individuals in the ELF parse via token index matching.

This Work: Aligning FN parses with EL parses

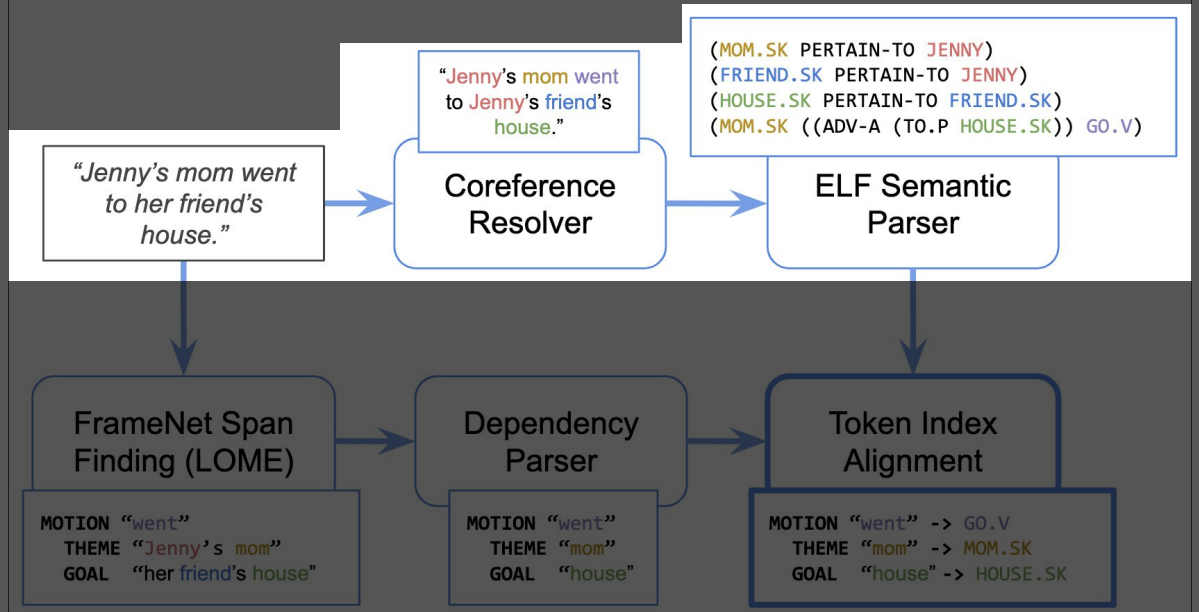


Figure 2: The architecture of the system. Raw story text is fed along two tracks: the logical-semantic parsing track, shown along the top, and the FrameNet parsing track, shown along the bottom. The FrameNet text spans are reduced to direct object tokens and correlated with logical individuals in the ELF parse via token index matching.

This Work: Aligning FN parses with EL parses

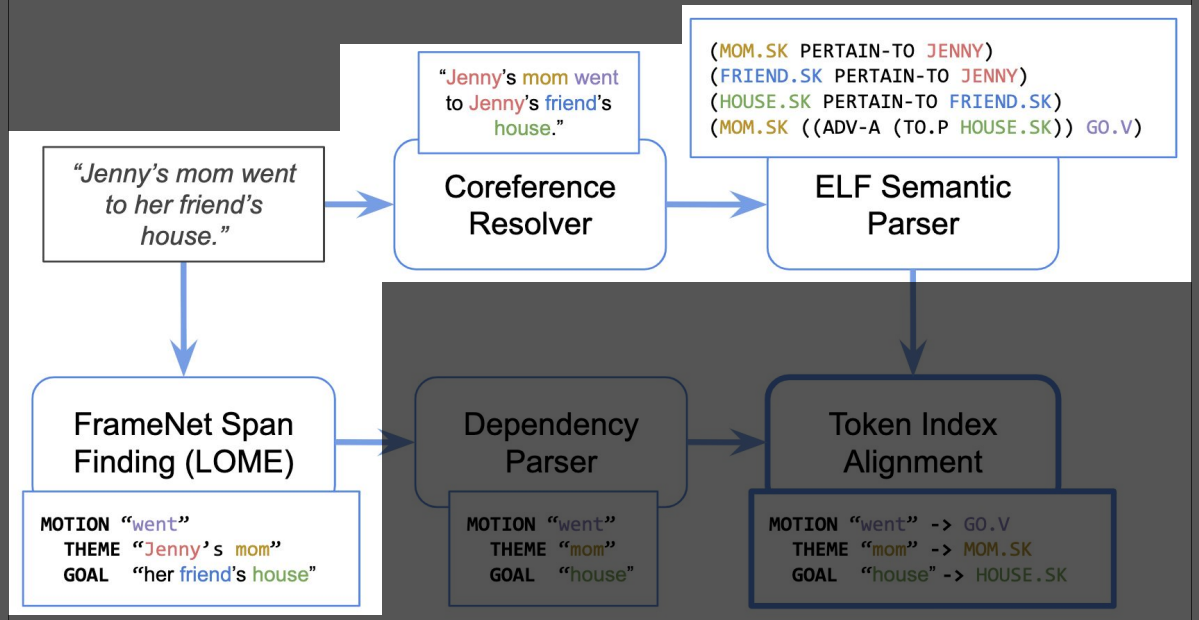


Figure 2: The architecture of the system. Raw story text is fed along two tracks: the logical-semantic parsing track, shown along the top, and the FrameNet parsing track, shown along the bottom. The FrameNet text spans are reduced to direct object tokens and correlated with logical individuals in the ELF parse via token index matching.

This Work: Aligning FN parses with EL parses

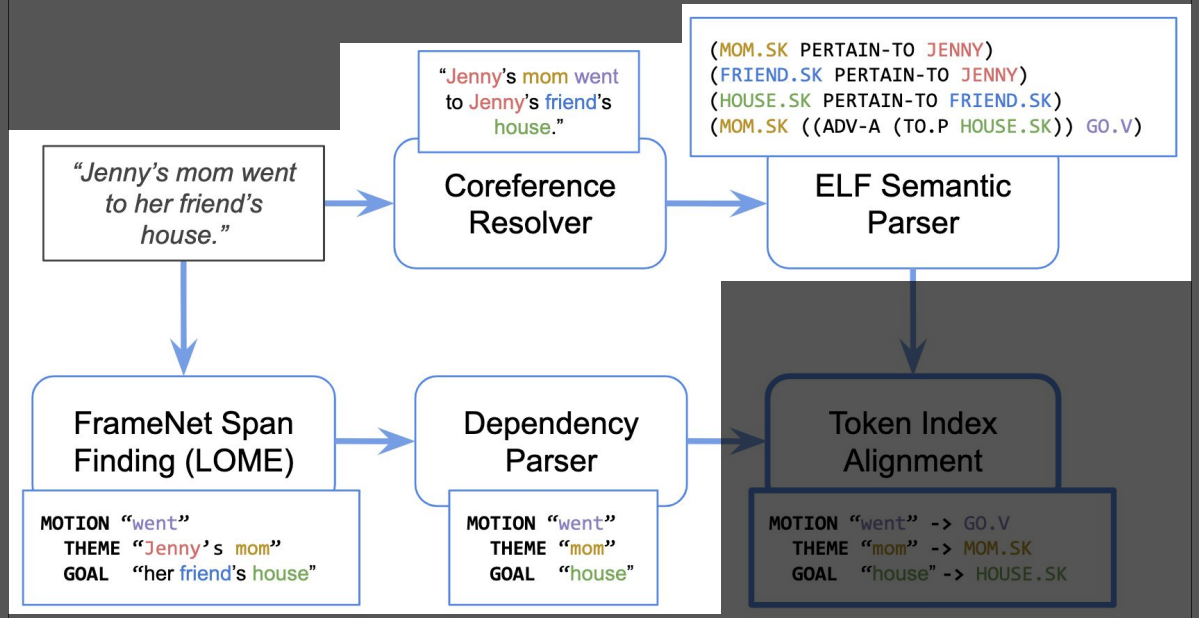


Figure 2: The architecture of the system. Raw story text is fed along two tracks: the logical-semantic parsing track, shown along the top, and the FrameNet parsing track, shown along the bottom. The FrameNet text spans are reduced to direct object tokens and correlated with logical individuals in the ELF parse via token index matching.

This Work: Aligning FN parses with EL parses

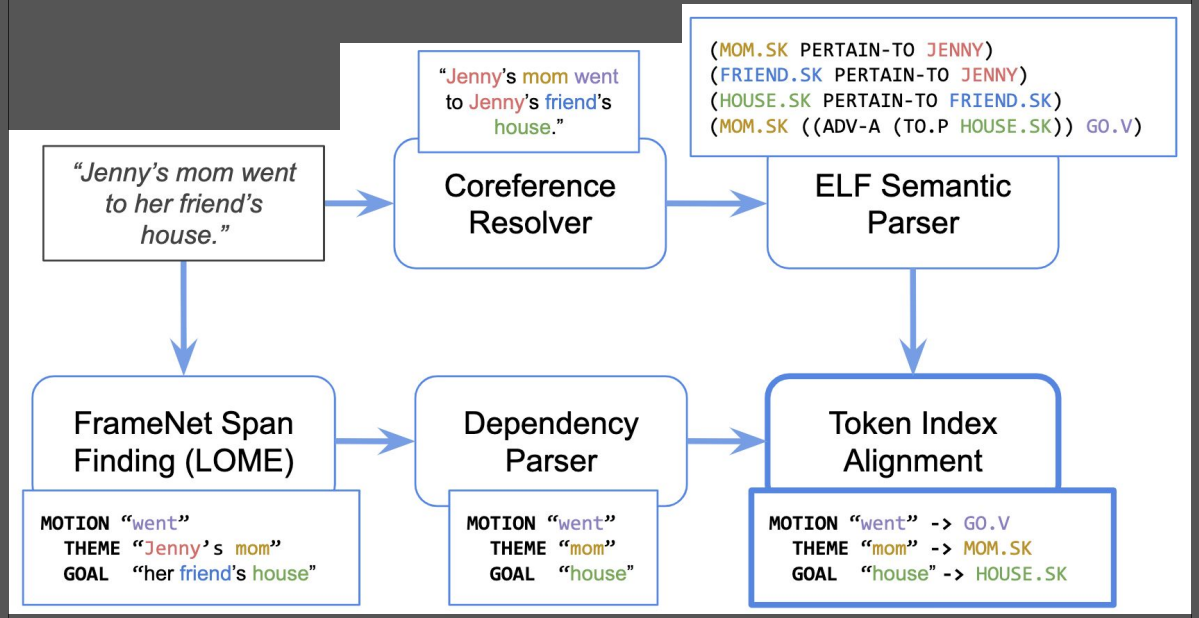


Figure 2: The architecture of the system. Raw story text is fed along two tracks: the logical-semantic parsing track, shown along the top, and the FrameNet parsing track, shown along the bottom. The FrameNet text spans are reduced to direct object tokens and correlated with logical individuals in the ELF parse via token index matching.

This Work: Aligning FN parses with EL parses

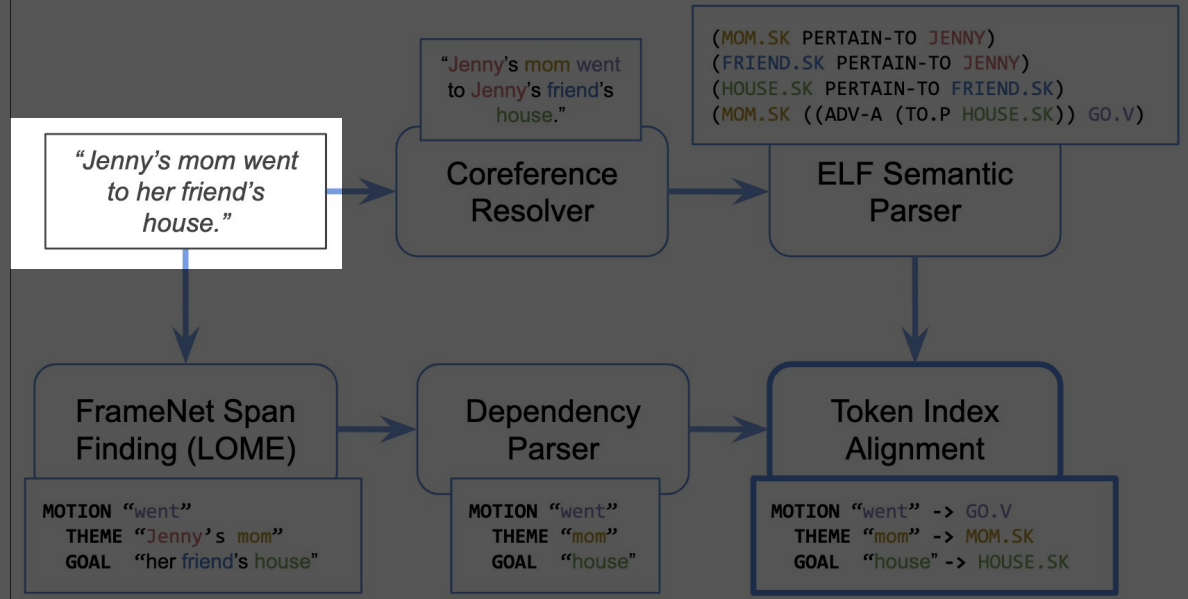


Figure 2: The architecture of the system. Raw story text is fed along two tracks: the logical-semantic parsing track, shown along the top, and the FrameNet parsing track, shown along the bottom. The FrameNet text spans are reduced to direct object tokens and correlated with logical individuals in the ELF parse via token index matching.

This Work: Aligning FN parses with EL parses

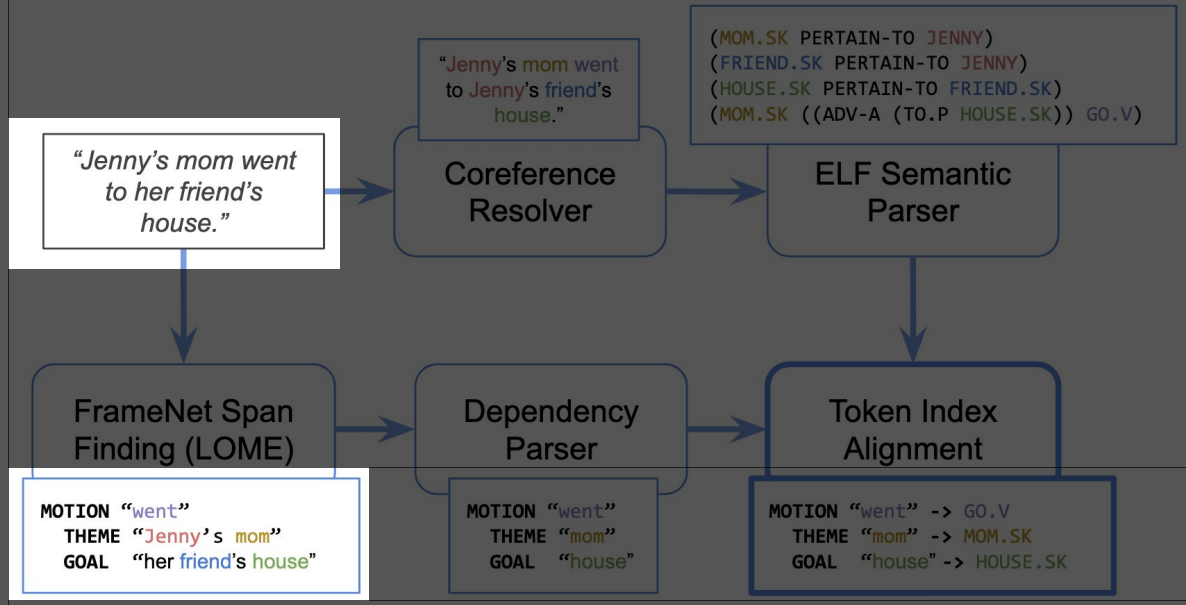


Figure 2: The architecture of the system. Raw story text is fed along two tracks: the logical-semantic parsing track, shown along the top, and the FrameNet parsing track, shown along the bottom. The FrameNet text spans are reduced to direct object tokens and correlated with logical individuals in the ELF parse via token index matching.

This Work: Aligning FN parses with EL parses

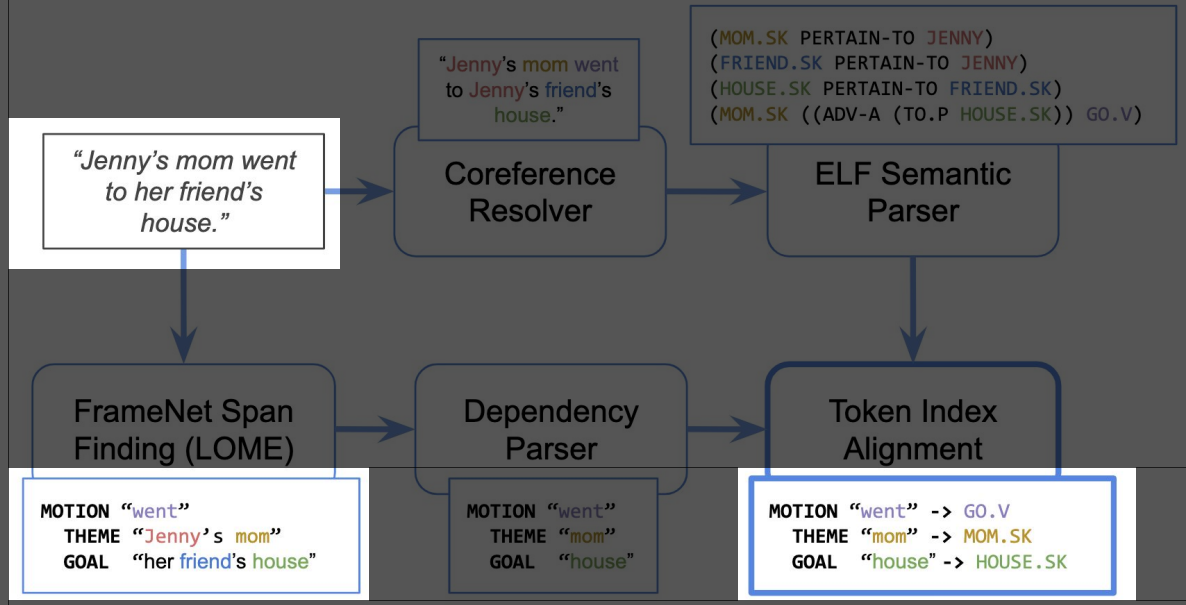


Figure 2: The architecture of the system. Raw story text is fed along two tracks: the logical-semantic parsing track, shown along the top, and the FrameNet parsing track, shown along the bottom. The FrameNet text spans are reduced to direct object tokens and correlated with logical individuals in the ELF parse via token index matching.

This Work: Aligning FN parses with EL parses

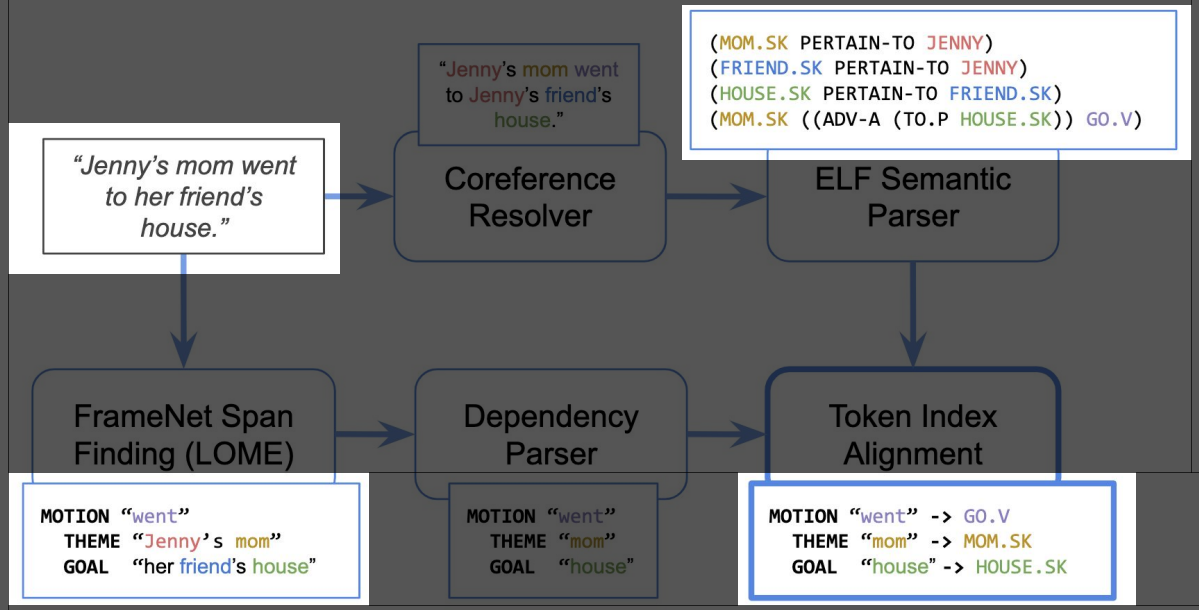


Figure 2: The architecture of the system. Raw story text is fed along two tracks: the logical-semantic parsing track, shown along the top, and the FrameNet parsing track, shown along the bottom. The FrameNet text spans are reduced to direct object tokens and correlated with logical individuals in the ELF parse via token index matching.

```
MOTION GO.V
  THEME MOM.SK
  GOAL  HOUSE.SK
```

```
MOTION GO.V
  THEME MOM.SK
  GOAL  HOUSE.SK
```

```
(MOM.SK ((ADV-A (TO.P HOUSE.SK)) GO.V)
```

```
MOTION GO.V
  THEME MOM.SK
  GOAL  HOUSE.SK
```

```
(MOM.SK ((ADV-A (TO.P HOUSE.SK)) GO.V)
```

```
(EPI-SCHEMA ((?X TRAVEL.V ?S ?D) ** ?E)
      :Roles
            !R1 (?X PERSON.N)
            !R2 (?D LOCATION.N)
            !R3 (?S LOCATION.N)

      :Goals
            ?G1 (?X WANT.V (THAT (?X AT.P ?D)))

      :Preconditions
            ?I1 (NOT (?X AT.P ?D))
            ?I2 (?X AT.P ?S)

      :Postconditions
            ?P1 (NOT (?X AT.P ?S))
            ?P2 (?X AT.P ?D)
)
```

```
MOTION GO.V
   THEME MOM.SK
   GOAL  HOUSE.SK
```

```
(motion * -> travel.v
                ((?x theme)
                        pre-arg
                        (theme (if not event))
                )
                ((?s source)
                        (adv from.p)
                        (source (if not event))
                        path
                )
                ((?d goal)
                        (adv to.p)
                        (post-arg (1 of any))
                        (goal (if not event))
                        path
                )
        )
```

```
(MOM.SK ((ADV-A (TO.P HOUSE.SK)) GO.V)
```

```
(EPI-SCHEMA ((?X TRAVEL.V ?S ?D) ** ?E)
        :Roles
                !R1 (?X PERSON.N)
                !R2 (?D LOCATION.N)
                !R3 (?S LOCATION.N)

        :Goals
                ?G1 (?X WANT.V (THAT (?X AT.P ?D)))

        :Preconditions
                ?I1 (NOT (?X AT.P ?D))
                ?I2 (?X AT.P ?S)

        :Postconditions
                ?P1 (NOT (?X AT.P ?S))
                ?P2 (?X AT.P ?D)
)
```

```
MOTION GO.V
   THEME MOM.SK
   GOAL  HOUSE.SK
```

```
(motion * -> travel.v
                ((?x theme)
                        pre-arg
                        (theme (if not event))
                )
                ((?s source)
                        (adv from.p)
                        (source (if not event))
                        path
                )
                ((?d goal)
                        (adv to.p)
                        (post-arg (1 of any))
                        (goal (if not event))
                        path
                )
        )
```

```
(MOM.SK ((ADV-A (TO.P HOUSE.SK)) GO.V)
```

```
(EPI-SCHEMA ((?X TRAVEL.V ?S ?D) ** ?E)
        :Roles
                !R1 (?X PERSON.N)
                !R2 (?D LOCATION.N)
                !R3 (?S LOCATION.N)

        :Goals
                ?G1 (?X WANT.V (THAT (?X AT.P ?D)))

        :Preconditions
                ?I1 (NOT (?X AT.P ?D))
                ?I2 (?X AT.P ?S)

        :Postconditions
                ?P1 (NOT (?X AT.P ?S))
                ?P2 (?X AT.P ?D)
)
```

```
MOTION GO.V
   THEME MOM.SK
   GOAL  HOUSE.SK
```

```
(motion * -> travel.v
                ((?x theme)
                        pre-arg
                        (theme (if not event))
                )
                ((?s source)
                        (adv from.p)
                        (source (if not event))
                        path
                )
                ((?d goal)
                        (adv to.p)
                        (post-arg (1 of any))
                        (goal (if not event))
                        path
                )
        )
```

```
(MOM.SK ((ADV-A (TO.P HOUSE.SK)) GO.V)
```

```
(EPI-SCHEMA ((?X TRAVEL.V ?S ?D) ** ?E)
        :Roles
                !R1 (?X PERSON.N)
                !R2 (?D LOCATION.N)
                !R3 (?S LOCATION.N)

        :Goals
                ?G1 (?X WANT.V (THAT (?X AT.P ?D)))

        :Preconditions
                ?I1 (NOT (?X AT.P ?D))
                ?I2 (?X AT.P ?S)

        :Postconditions
                ?P1 (NOT (?X AT.P ?S))
                ?P2 (?X AT.P ?D)
)
```

```
MOTION GO.V
   THEME MOM.SK
   GOAL  HOUSE.SK
```

```
(motion * -> travel.v
                ((?x theme)
                        pre-arg
                        (theme (if not event))
                )
                ((?s source)
                        (adv from.p)
                        (source (if not event))
                        path
                )
                ((?d goal)
                        (adv to.p)
                        (post-arg (1 of any))
                        (goal (if not event))
                        path
                )
        )
```

```
(MOM.SK ((ADV-A (TO.P HOUSE.SK)) GO.V)
```

```
(EPI-SCHEMA ((?X TRAVEL.V ?S ?D) ** ?E)
        :Roles
                !R1 (?X PERSON.N)
                !R2 (?D LOCATION.N)
                !R3 (?S LOCATION.N)

        :Goals
                ?G1 (?X WANT.V (THAT (?X AT.P ?D)))

        :Preconditions
                ?I1 (NOT (?X AT.P ?D))
                ?I2 (?X AT.P ?S)

        :Postconditions
                ?P1 (NOT (?X AT.P ?S))
                ?P2 (?X AT.P ?D)
)
```

```
MOTION GO.V
   THEME MOM.SK
   GOAL  HOUSE.SK
```

```
(motion * -> travel.v
                ((?x theme)
                        pre-arg
                        (theme (if not event))
                )
                ((?s source)
                        (adv from.p)
                        (source (if not event))
                        path
                )
                ((?d goal)
                        (adv to.p)
                        (post-arg (1 of any))
                        (goal (if not event))
                        path
                )
        )
```

```
(MOM.SK ((ADV-A (TO.P HOUSE.SK)) GO.V)
```

```
(EPI-SCHEMA ((?X TRAVEL.V ?S ?D) ** ?E)
        :Roles
                !R1 (?X PERSON.N)
                !R2 (?D LOCATION.N)
                !R3 (?S LOCATION.N)

        :Goals
                ?G1 (?X WANT.V (THAT (?X AT.P ?D)))

        :Preconditions
                ?I1 (NOT (?X AT.P ?D))
                ?I2 (?X AT.P ?S)

        :Postconditions
                ?P1 (NOT (?X AT.P ?S))
                ?P2 (?X AT.P ?D)
)
```

```
MOTION GO.V
   THEME MOM.SK
   GOAL  HOUSE.SK
```

```
(motion * -> travel.v
                ((?x theme)
                        pre-arg
                        (theme (if not event))
                )
                ((?s source)
                        (adv from.p)
                        (source (if not event))
                        path
                )
                ((?d goal)
                        (adv to.p)
                        (post-arg (1 of any))
                        (goal (if not event))
                        path
                )
        )
```

```
(MOM.SK ((ADV-A (TO.P HOUSE.SK)) GO.V)
```

```
(EPI-SCHEMA ((?X TRAVEL.V ?S ?D) ** ?E)
        :Roles
                !R1 (?X PERSON.N)
                !R2 (?D LOCATION.N)
                !R3 (?S LOCATION.N)

        :Goals
                ?G1 (?X WANT.V (THAT (?X AT.P ?D)))

        :Preconditions
                ?I1 (NOT (?X AT.P ?D))
                ?I2 (?X AT.P ?S)

        :Postconditions
                ?P1 (NOT (?X AT.P ?S))
                ?P2 (?X AT.P ?D)
)
```

```
MOTION GO.V
   THEME MOM.SK
   GOAL  HOUSE.SK
```

```
(motion * -> travel.v
                ((?x theme)
                        pre-arg
                        (theme (if not event))
                )
                ((?s source)
                        (adv from.p)
                        (source (if not event))
                        path
                )
                ((?d goal)
                        (adv to.p)
                        (post-arg (1 of any))
                        (goal (if not event))
                        path
                )
        )
```

```
(MOM.SK ((ADV-A (TO.P HOUSE.SK)) GO.V)
```

```
(EPI-SCHEMA ((?X TRAVEL.V ?S ?D) ** ?E)
        :Roles
                !R1 (?X PERSON.N)
                !R2 (?D LOCATION.N)
                !R3 (?S LOCATION.N)

        :Goals
                ?G1 (?X WANT.V (THAT (?X AT.P ?D)))

        :Preconditions
                ?I1 (NOT (?X AT.P ?D))
                ?I2 (?X AT.P ?S)

        :Postconditions
                ?P1 (NOT (?X AT.P ?S))
                ?P2 (?X AT.P ?D)
)
```

```
MOTION GO.V
   THEME MOM.SK
   GOAL  HOUSE.SK
```

```
(motion * -> travel.v
                ((?x theme)
                        pre-arg
                        (theme (if not event))
                )
                ((?s source)
                        (adv from.p)
                        (source (if not event))
                        path
                )
                ((?d goal)
                        (adv to.p)
                        (post-arg (1 of any))
                        (goal (if not event))
                        path
                )
        )
```

```
(MOM.SK ((ADV-A (TO.P HOUSE.SK)) GO.V)
```

```
(EPI-SCHEMA ((?X TRAVEL.V ?S ?D) ** ?E)
        :Roles
                !R1 (?X PERSON.N)
                !R2 (?D LOCATION.N)
                !R3 (?S LOCATION.N)

        :Goals
                ?G1 (?X WANT.V (THAT (?X AT.P ?D)))

        :Preconditions
                ?I1 (NOT (?X AT.P ?D))
                ?I2 (?X AT.P ?S)

        :Postconditions
                ?P1 (NOT (?X AT.P ?S))
                ?P2 (?X AT.P ?D)
)
```

```
MOTION GO.V
   THEME MOM.SK
   GOAL  HOUSE.SK
```

```
(motion * -> travel.v
                ((?x theme)
                        pre-arg
                        (theme (if not event))
                )
                ((?s source)
                        (adv from.p)
                        (source (if not event))
                        path
                )
                ((?d goal)
                        (adv to.p)
                        (post-arg (1 of any))
                        (goal (if not event))
                        path
                )
        )
```

```
(MOM.SK ((ADV-A (TO.P HOUSE.SK)) GO.V)
```

```
(EPI-SCHEMA ((?X TRAVEL.V ?S ?D) ** ?E)
        :Roles
                !R1 (?X PERSON.N)
                !R2 (?D LOCATION.N)
                !R3 (?S LOCATION.N)

        :Goals
                ?G1 (?X WANT.V (THAT (?X AT.P ?D)))

        :Preconditions
                ?I1 (NOT (?X AT.P ?D))
                ?I2 (?X AT.P ?S)

        :Postconditions
                ?P1 (NOT (?X AT.P ?S))
                ?P2 (?X AT.P ?D)
)
```

```
MOTION GO.V
   THEME MOM.SK
   GOAL  HOUSE.SK
```

```
(motion * -> travel.v
                ((?x theme)
                        pre-arg
                        (theme (if not event))
                )
                ((?s source)
                        (adv from.p)
                        (source (if not event))
                        path
                )
                ((?d goal)
                        (adv to.p)
                        (post-arg (1 of any))
                        (goal (if not event))
                        path
                )
        )
```

```
(MOM.SK ((ADV-A (TO.P HOUSE.SK)) GO.V)
```

```
(EPI-SCHEMA ((?X TRAVEL.V ?S ?D) ** ?E)
        :Roles
                !R1 (?X PERSON.N)
                !R2 (?D LOCATION.N)
                !R3 (?S LOCATION.N)

        :Goals
                ?G1 (?X WANT.V (THAT (?X AT.P ?D)))

        :Preconditions
                ?I1 (NOT (?X AT.P ?D))
                ?I2 (?X AT.P ?S)

        :Postconditions
                ?P1 (NOT (?X AT.P ?S))
                ?P2 (?X AT.P ?D)
)
```

```
MOTION GO.V
   THEME MOM.SK
   GOAL  HOUSE.SK
```

```
(motion * -> travel.v
                ((?x theme)
                        pre-arg
                        (theme (if not event))
                )
                ((?s source)
                        (adv from.p)
                        (source (if not event))
                        path
                )
                ((?d goal)
                        (adv to.p)
                        (post-arg (1 of any))
                        (goal (if not event))
                        path
                )
        )
```

```
(MOM.SK ((ADV-A (TO.P HOUSE.SK)) GO.V)
```

```
(EPI-SCHEMA ((?X TRAVEL.V ?S ?D) ** ?E)
        :Roles
                !R1 (MOM.SK PERSON.N)
                !R2 (HOUSE.SK LOCATION.N)
                !R3 (?S LOCATION.N)

        :Goals
                ?G1 (?X WANT.V (THAT (?X AT.P ?D)))

        :Preconditions
                ?I1 (NOT (?X AT.P ?D))
                ?I2 (?X AT.P ?S)

        :Postconditions
                ?P1 (NOT (?X AT.P ?S))
                ?P2 (?X AT.P ?D)
)
```

```
MOTION GO.V
   THEME MOM.SK
   GOAL  HOUSE.SK
```

```
(motion * -> travel.v
                ((?x theme)
                        pre-arg
                        (theme (if not event))
                )
                ((?s source)
                        (adv from.p)
                        (source (if not event))
                        path
                )
                ((?d goal)
                        (adv to.p)
                        (post-arg (1 of any))
                        (goal (if not event))
                        path
                )
        )
```

```
(MOM.SK ((ADV-A (TO.P HOUSE.SK)) GO.V)
```

```
(EPI-SCHEMA ((MOM.SK TRAVEL.V ?S HOUSE.SK) ** ?E)
        :Roles
                !R1 (MOM.SK PERSON.N)
                !R2 (HOUSE.SK LOCATION.N)
                !R3 (?S LOCATION.N)

        :Goals
             ?G1 (MOM.SK WANT.V (THAT (MOM.SK AT.P HOUSE.SK)))

        :Preconditions
                ?I1 (NOT (MOM.SK AT.P HOUSE.SK))
                ?I2 (MOM.SK AT.P ?S)

        :Postconditions
                ?P1 (NOT (MOM.SK AT.P ?S))
                ?P2 (MOM.SK AT.P HOUSE.SK)
)
```

```
(EPI-SCHEMA ((MOM.SK TRAVEL.V ?S HOUSE.SK) ** ?E)
      :Roles
            !R1 (MOM.SK PERSON.N)
            !R2 (HOUSE.SK LOCATION.N)
            !R3 (?S LOCATION.N)

      :Goals
           ?G1 (MOM.SK WANT.V (THAT (MOM.SK AT.P HOUSE.SK)))

      :Preconditions
            ?I1 (NOT (MOM.SK AT.P HOUSE.SK))
            ?I2 (MOM.SK AT.P ?S)

      :Postconditions
            ?P1 (NOT (MOM.SK AT.P ?S))
            ?P2 (MOM.SK AT.P HOUSE.SK)
)
```

```
(MOM.SK PERTAIN-TO JENNY)
(FRIEND.SK PERTAIN-TO JENNY)
(HOUSE.SK PERTAIN-TO FRIEND.SK)
(MOM.SK ((ADV-A (TO.P HOUSE.SK)) GO.V)
```

```
(EPI-SCHEMA ((MOM.SK TRAVEL.V ?S HOUSE.SK) ** ?E)
      :Roles
            !R1 (MOM.SK PERSON.N)
            !R2 (HOUSE.SK LOCATION.N)
            !R3 (?S LOCATION.N)

      :Goals
           ?G1 (MOM.SK WANT.V (THAT (MOM.SK AT.P HOUSE.SK)))

      :Preconditions
            ?I1 (NOT (MOM.SK AT.P HOUSE.SK))
            ?I2 (MOM.SK AT.P ?S)

      :Postconditions
            ?P1 (NOT (MOM.SK AT.P ?S))
            ?P2 (MOM.SK AT.P HOUSE.SK)
)
```

```
(MOM.SK PERTAIN-TO JENNY)
(FRIEND.SK PERTAIN-TO JENNY)
(HOUSE.SK PERTAIN-TO FRIEND.SK)
(MOM.SK ((ADV-A (TO.P HOUSE.SK)) GO.V)

(MOM.SK MOM.N)
(FRIEND.SK FRIEND.N)
(HOUSE.SK HOUSE.N)
```

```
(EPI-SCHEMA ((MOM.SK TRAVEL.V ?S HOUSE.SK) ** ?E)
      :Roles
            !R1 (MOM.SK PERSON.N)
            !R2 (HOUSE.SK LOCATION.N)
            !R3 (?S LOCATION.N)

      :Goals
           ?G1 (MOM.SK WANT.V (THAT (MOM.SK AT.P HOUSE.SK)))

      :Preconditions
            ?I1 (NOT (MOM.SK AT.P HOUSE.SK))
            ?I2 (MOM.SK AT.P ?S)

      :Postconditions
            ?P1 (NOT (MOM.SK AT.P ?S))
            ?P2 (MOM.SK AT.P HOUSE.SK)
)
```

```
(MOM.SK PERTAIN-TO JENNY)
(FRIEND.SK PERTAIN-TO JENNY)
(HOUSE.SK PERTAIN-TO FRIEND.SK)
(MOM.SK ((ADV-A (TO.P HOUSE.SK)) GO.V)

(MOM.SK MOM.N)
(FRIEND.SK FRIEND.N)
(HOUSE.SK HOUSE.N)
```

```
(EPI-SCHEMA ((MOM.SK TRAVEL.V ?S HOUSE.SK) ** ?E)
      :Roles
            !R1 (MOM.SK PERSON.N)
            !R2 (HOUSE.SK LOCATION.N)
            !R3 (?S LOCATION.N)
            !R4 (MOM.SK MOM.N)
            !R5 (HOUSE.SK HOUSE.N)
            !R6 (MOM.SK PERTAIN-TO JENNY)
            !R7 (HOUSE.SK PERTAIN-TO FRIEND.SK)
            !R8 (FRIEND.SK FRIEND.N)
            !R9 (FRIEND.SK PERTAIN-TO JENNY)

      :Goals
           ?G1 (MOM.SK WANT.V (THAT (MOM.SK AT.P HOUSE.SK)))

      :Preconditions
            ?I1 (NOT (MOM.SK AT.P HOUSE.SK))
            ?I2 (MOM.SK AT.P ?S)

      :Postconditions
            ?P1 (NOT (MOM.SK AT.P ?S))
            ?P2 (MOM.SK AT.P HOUSE.SK)
)
```

```
(MOM.SK PERTAIN-TO JENNY)
(FRIEND.SK PERTAIN-TO JENNY)
(HOUSE.SK PERTAIN-TO FRIEND.SK)
(MOM.SK ((ADV-A (TO.P HOUSE.SK)) GO.V)

(MOM.SK MOM.N)
(FRIEND.SK FRIEND.N)
(HOUSE.SK HOUSE.N)
```

```
(EPI-SCHEMA ((MOM.SK TRAVEL.V ?S HOUSE.SK) ** ?E)
      :Roles
            !R1 (MOM.SK PERSON.N)
            !R2 (HOUSE.SK LOCATION.N)
            !R3 (?S LOCATION.N)
            !R4 (MOM.SK MOM.N)
            !R5 (HOUSE.SK HOUSE.N)
            !R6 (MOM.SK PERTAIN-TO JENNY)
            !R7 (HOUSE.SK PERTAIN-TO FRIEND.SK)
            !R8 (FRIEND.SK FRIEND.N)
            !R9 (FRIEND.SK PERTAIN-TO JENNY)

      :Goals
           ?G1 (MOM.SK WANT.V (THAT (MOM.SK AT.P HOUSE.SK)))

      :Preconditions
            ?I1 (NOT (MOM.SK AT.P HOUSE.SK))
            ?I2 (MOM.SK AT.P ?S)

      :Postconditions
            ?P1 (NOT (MOM.SK AT.P ?S))
            ?P2 (MOM.SK AT.P HOUSE.SK)
)
```

```
(EPI-SCHEMA ((MOM.SK TRAVEL.V ?S HOUSE.SK) ** ?E)
      :Roles
            !R1 (MOM.SK PERSON.N)
            !R2 (HOUSE.SK LOCATION.N)
            !R3 (?S LOCATION.N)
            !R4 (MOM.SK MOM.N)
            !R5 (HOUSE.SK HOUSE.N)
            !R6 (MOM.SK PERTAIN-TO JENNY)
            !R7 (HOUSE.SK PERTAIN-TO FRIEND.SK)
            !R8 (FRIEND.SK FRIEND.N)
            !R9 (FRIEND.SK PERTAIN-TO JENNY)

      :Goals
           ?G1 (MOM.SK WANT.V (THAT (MOM.SK AT.P HOUSE.SK)))

      :Preconditions
            ?I1 (NOT (MOM.SK AT.P HOUSE.SK))
            ?I2 (MOM.SK AT.P ?S)

      :Postconditions
            ?P1 (NOT (MOM.SK AT.P ?S))
            ?P2 (MOM.SK AT.P HOUSE.SK)
)
```

```
(EPI-SCHEMA ((?X TRAVEL.V ?S ?D) ** ?E)
      :Roles
            !R1 (?X PERSON.N)
            !R2 (?D LOCATION.N)
            !R3 (?S LOCATION.N)
            !R4 (?X MOM.N)
            !R5 (?D HOUSE.N)
            !R6 (?X PERTAIN-TO ?Y)
            !R7 (?D PERTAIN-TO ?F)
            !R8 (?F FRIEND.N)
            !R9 (?F PERTAIN-TO ?Y)

      :Goals
           ?G1 (?X WANT.V (THAT (?X AT.P ?D)))

      :Preconditions
            ?I1 (NOT (?X AT.P ?D))
            ?I2 (?X AT.P ?S)

      :Postconditions
            ?P1 (NOT (?X AT.P ?S))
            ?P2 (?X AT.P ?D)
)
```

"Jenny's mom went to Jenny's friend's house."

```
(EPI-SCHEMA ((?X TRAVEL.V ?S ?D) ** ?E)
      :Roles
            !R1 (?X PERSON.N)
            !R2 (?D LOCATION.N)
            !R3 (?S LOCATION.N)
            !R4 (?X MOM.N)
            !R5 (?D HOUSE.N)
            !R6 (?X PERTAIN-TO ?Y)
            !R7 (?D PERTAIN-TO ?F)
            !R8 (?F FRIEND.N)
            !R9 (?F PERTAIN-TO ?Y)

      :Goals
           ?G1 (?X WANT.V (THAT (?X AT.P ?D)))

      :Preconditions
            ?I1 (NOT (?X AT.P ?D))
            ?I2 (?X AT.P ?S)

      :Postconditions
            ?P1 (NOT (?X AT.P ?S))
            ?P2 (?X AT.P ?D)
)
```

"Jenny's mom went to Jenny's friend's house."

LOME

```
MOTION "went"
   THEME "Jenny's mom"
   GOAL  "her friend's house"
```

EL alignment

```
MOTION GO.V
   THEME MOM.SK
   GOAL  HOUSE.SK
```

Protoschema mapping

```
(EPI-SCHEMA ((?X TRAVEL.V ?S ?D) ** ?E)
      :Roles
            !R1 (?X PERSON.N)
            !R2 (?D LOCATION.N)
            !R3 (?S LOCATION.N)
            !R4 (?X MOM.N)
            !R5 (?D HOUSE.N)
                (?X PERTAIN-TO ?Y)
                (?D PERTAIN-TO ?F)
                (?F FRIEND.N)
                (?F PERTAIN-TO ?Y)

            ?G1 (?X WANT.V (THAT (?X AT.P ?D)))

      :Preconditions
            ?T1 (NOT (?X AT.P ?D))

                                  ?S))
)
```

"Jenny's mom went to Jenny's friend's house."

```
(EPI-SCHEMA ((?X TRAVEL.V ?S ?D) ** ?E)
      :Roles
            !R1 (?X PERSON.N)
            !R2 (?D LOCATION.N)
            !R3 (?S LOCATION.N)
            !R4 (?X MOM.N)
            !R5 (?D HOUSE.N)
            !R6 (?X PERTAIN-TO ?Y)
            !R7 (?D PERTAIN-TO ?F)
            !R8 (?F FRIEND.N)
            !R9 (?F PERTAIN-TO ?Y)

      :Goals
           ?G1 (?X WANT.V (THAT (?X AT.P ?D)))

      :Preconditions
            ?I1 (NOT (?X AT.P ?D))
            ?I2 (?X AT.P ?S)

      :Postconditions
            ?P1 (NOT (?X AT.P ?S))
            ?P2 (?X AT.P ?D)
)
```

"Jenny's mom went to Jenny's friend's house."

```
(EPI-SCHEMA ((?X TRAVEL.V ?S ?D) ** ?E)
      :Roles
            !R1 (?X PERSON.N)
            !R2 (?D LOCATION.N)
            !R3 (?S LOCATION.N)
            !R4 (?X MOM.N)
            !R5 (?D HOUSE.N)
            !R6 (?X PERTAIN-TO ?Y)
            !R7 (?D PERTAIN-TO ?F)
            !R8 (?F FRIEND.N)
            !R9 (?F PERTAIN-TO ?Y)

      :Goals
            ?G1 (?X WANT.V (THAT (?X AT.P ?D)))

      :Preconditions
            ?I1 (NOT (?X AT.P ?D))
            ?I2 (?X AT.P ?S)

      :Postconditions
            ?P1 (NOT (?X AT.P ?S))
            ?P2 (?X AT.P ?D)
)
```

"Jenny's mom went to Jenny's friend's house."

```
(EPI-SCHEMA ((?X TRAVEL.V ?S ?D) ** ?E)
      :Roles
            !R1 (?X PERSON.N)
            !R2 (?D LOCATION.N)
            !R3 (?S LOCATION.N)
            !R4 (?X MOM.N)
            !R5 (?D HOUSE.N)
            !R6 (?X PERTAIN-TO ?Y)
            !R7 (?D PERTAIN-TO ?F)
            !R8 (?F FRIEND.N)
            !R9 (?F PERTAIN-TO ?Y)

      :Goals
           ?G1 (?X WANT.V (THAT (?X AT.P ?D)))

      :Preconditions
            ?I1 (NOT (?X AT.P ?D))
            ?I2 (?X AT.P ?S)

      :Postconditions
            ?P1 (NOT (?X AT.P ?S))
            ?P2 (?X AT.P ?D)
)
```

"Jenny's mom went to Jenny's friend's house."

"Maria went to Jeff's friend's house."

```
(EPI-SCHEMA ((?X TRAVEL.V ?S ?D) ** ?E)
      :Roles
            !R1 (?X PERSON.N)
            !R2 (?D LOCATION.N)
            !R3 (?S LOCATION.N)
            !R4 (?X MOM.N)
            !R5 (?D HOUSE.N)
            !R6 (?X PERTAIN-TO ?Y)
            !R7 (?D PERTAIN-TO ?F)
            !R8 (?F FRIEND.N)
            !R9 (?F PERTAIN-TO ?Y)

      :Goals
           ?G1 (?X WANT.V (THAT (?X AT.P ?D)))

      :Preconditions
            ?I1 (NOT (?X AT.P ?D))
            ?I2 (?X AT.P ?S)

      :Postconditions
            ?P1 (NOT (?X AT.P ?S))
            ?P2 (?X AT.P ?D)
)
```

"Jenny's mom went to Jenny's friend's house."

"Maria went to Jeff's friend's house."

```
(EPI-SCHEMA ((?X TRAVEL.V ?S ?D) ** ?E)
      :Roles
            !R1 (?X PERSON.N)
            !R2 (?D LOCATION.N)
            !R3 (?S LOCATION.N)
            !R4 (?X MOM.N)
            !R5 (?D HOUSE.N)
            !R6 (?X PERTAIN-TO ?Y)
            !R7 (?D PERTAIN-TO ?F)
            !R8 (?F FRIEND.N)
            !R9 (?F PERTAIN-TO ?Y)

      :Goals
           ?G1 (?X WANT.V (THAT (?X AT.P ?D)))

      :Preconditions
            ?I1 (NOT (?X AT.P ?D))
            ?I2 (?X AT.P ?S)

      :Postconditions
            ?P1 (NOT (?X AT.P ?S))
            ?P2 (?X AT.P ?D)
)
```

"Jenny's mom went to Jenny's friend's house."

"Maria went to Jeff's friend's house."

```
(EPI-SCHEMA ((MARIA TRAVEL.V ?S ?D) ** ?E)
      :Roles
            !R1 (MARIA PERSON.N)
            !R2 (?D LOCATION.N)
            !R3 (?S LOCATION.N)
            !R4 (MARIA MOM.N)
            !R5 (?D HOUSE.N)
            !R6 (MARIA PERTAIN-TO JEFF)
            !R7 (?D PERTAIN-TO ?F)
            !R8 (?F FRIEND.N)
            !R9 (?F PERTAIN-TO JEFF)

      :Goals
           ?G1 (MARIA WANT.V (THAT (MARIA AT.P ?D)))

      :Preconditions
            ?I1 (NOT (MARIA AT.P ?D))
            ?I2 (MARIA AT.P ?S)

      :Postconditions
            ?P1 (NOT (MARIA AT.P ?S))
            ?P2 (MARIA AT.P ?D)
)
```

"Jenny's mom went to Jenny's friend's house."

"Maria went to Jeff's friend's house."

```
(EPI-SCHEMA ((MARIA TRAVEL.V ?S ?D) ** ?E)
      :Roles
            !R1 (MARIA PERSON.N)
            !R2 (?D LOCATION.N)
            !R3 (?S LOCATION.N)
            !R4 (MARIA MOM.N)
            !R5 (?D HOUSE.N)
            !R6 (MARIA PERTAIN-TO JEFF)
            !R7 (?D PERTAIN-TO ?F)
            !R8 (?F FRIEND.N)
            !R9 (?F PERTAIN-TO JEFF)

      :Goals
           ?G1 (MARIA WANT.V (THAT (MARIA AT.P ?D)))

      :Preconditions
            ?I1 (NOT (MARIA AT.P ?D))
            ?I2 (MARIA AT.P ?S)

      :Postconditions
            ?P1 (NOT (MARIA AT.P ?S))
            ?P2 (MARIA AT.P ?D)
)
```

Info on Wider Schema Learning Project

Mining Logical Event Schemas From Pre-Trained Language Models

Lane Lawley, Lenhart Schubert

https://aclanthology.org/2022.acl-srw.25/

Thank you!