Lyna Beraich CMSC 691

Automated Narrative Planning Model Extension

Julie Porteous, João F. Ferreira,
Alan Lindsay, Marc Cavazza

What is Interactive Narrative?

- Automated storytelling where characters' actions are generated automatically
- Uses AI planning instead of branching story paths
- Challenges:
  - Must respond to user choices (robustness)
  - Must tell new, interesting stories on replay (diversity)
- Currently, humans must manually write all narrative actions which is too time consuming

The Core Idea

- Problem: domain solution is a bottleneck
- Solution: Automatically expand a small, human-made narrative using:
  - Anton - generates opposite actions
  - Thype - generates new types based on hierarchy

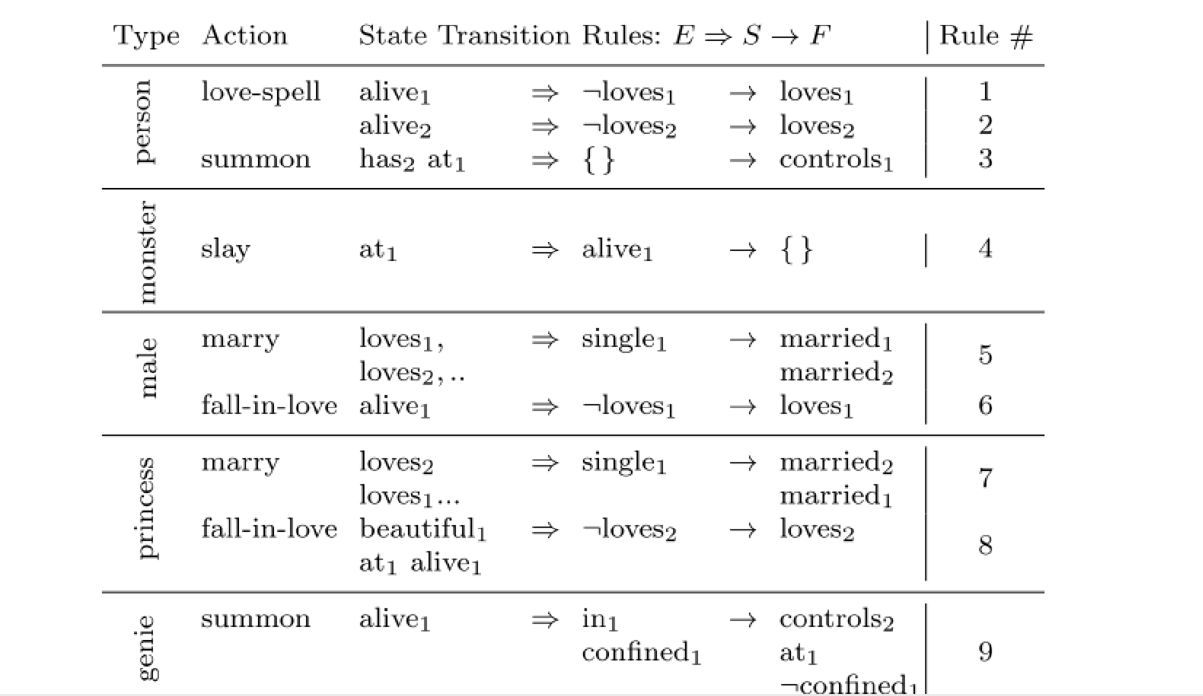
How does “anton” work?

- Analyzes existing actions to detect missing *reverse* transitions
- Uses linguistic resources to find antonyms
- Generates human-readable new actions
- Example:
  - love-spell
  - Anton generates: hate-spell
- Enables story recovery when the user disrupts the plan

How does “thype” work?

- Looks at type hierarchies (like “person → king → emperor”)
- Uses WordNet and ConceptNet to generate related types (hyponyms/hypernyms)
- Adds new types with minimal human input
- Enables variety by introducing new story elements.

Example State Transition Rules for the Aladdin Domain.

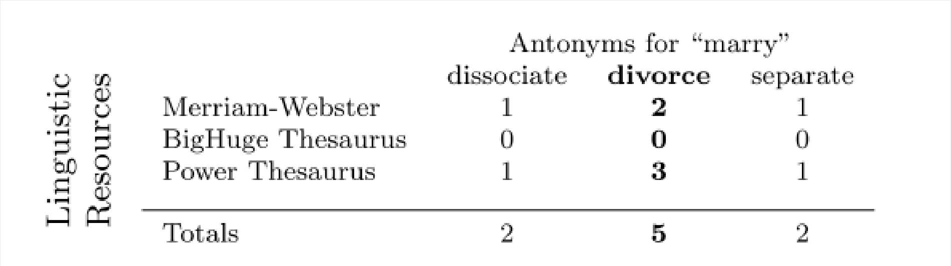


Type	Action	State Transition Rules: $E \Rightarrow S \rightarrow F$					Rule #
person	love-spell	alive$_1$	$\Rightarrow$	$\neg$loves$_1$	$\rightarrow$	loves$_1$	1
		alive$_2$	$\Rightarrow$	$\neg$loves$_2$	$\rightarrow$	loves$_2$	2
	summon	has$_2$ at$_1$	$\Rightarrow$	$\{\}$	$\rightarrow$	controls$_1$	3
monster	slay	at$_1$	$\Rightarrow$	alive$_1$	$\rightarrow$	$\{\}$	4
male	marry	loves$_1$, loves$_2$, ..	$\Rightarrow$	single$_1$	$\rightarrow$	married$_1$ married$_2$	5
	fall-in-love	alive$_1$	$\Rightarrow$	$\neg$loves$_1$	$\rightarrow$	loves$_1$	6
princess	marry	loves$_2$ loves$_1$...	$\Rightarrow$	single$_1$	$\rightarrow$	married$_2$ married$_1$	7
	fall-in-love	beautiful$_1$ at$_1$ alive$_1$	$\Rightarrow$	$\neg$loves$_2$	$\rightarrow$	loves$_2$	8
genie	summon	alive$_1$	$\Rightarrow$	in$_1$ confined$_1$	$\rightarrow$	controls$_2$ at$_1$ $\neg$confined$_1$	9

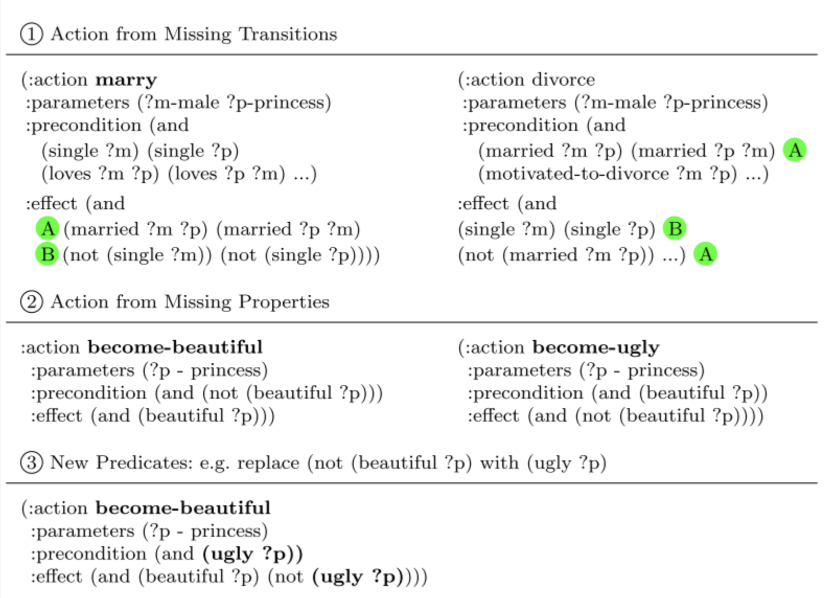
- Rules take the form $E \Rightarrow S \rightarrow F$ where:
  - E is a set of properties that enable the transition
  - S are the properties given up by the transition
  - F are properties acquired by the transition.
- The negative of a property x, used for analysing negative preconditions, is shown as ¬x.

Anton Process

- Antonym selection for the action marry
- The weights for each antonym are listed
- They are summed and the highest ranked is selected.
- In this case divorce is selected



Linguistic Resources	Antonyms for "marry" dissociate	 **divorce**	 separate
Merriam-Webster	1	**2**	1
BigHuge Thesaurus	0	**0**	0
Power Thesaurus	1	**3**	1
Totals	2	**5**	2



① Action from Missing Transitions

```
(:action marry
 :parameters (?m-male ?p-princess)
 :precondition (and
   (single ?m) (single ?p)
   (loves ?m ?p) (loves ?p ?m) ...)
 :effect (and
  A (married ?m ?p) (married ?p ?m)
  B (not (single ?m)) (not (single ?p))))
```

```
(:action divorce
 :parameters (?m-male ?p-princess)
 :precondition (and
   (married ?m ?p) (married ?p ?m) A
   (motivated-to-divorce ?m ?p) ...)
:effect (and
(single ?m) (single ?p) B
(not (married ?m ?p)) ...) A
```

② Action from Missing Properties

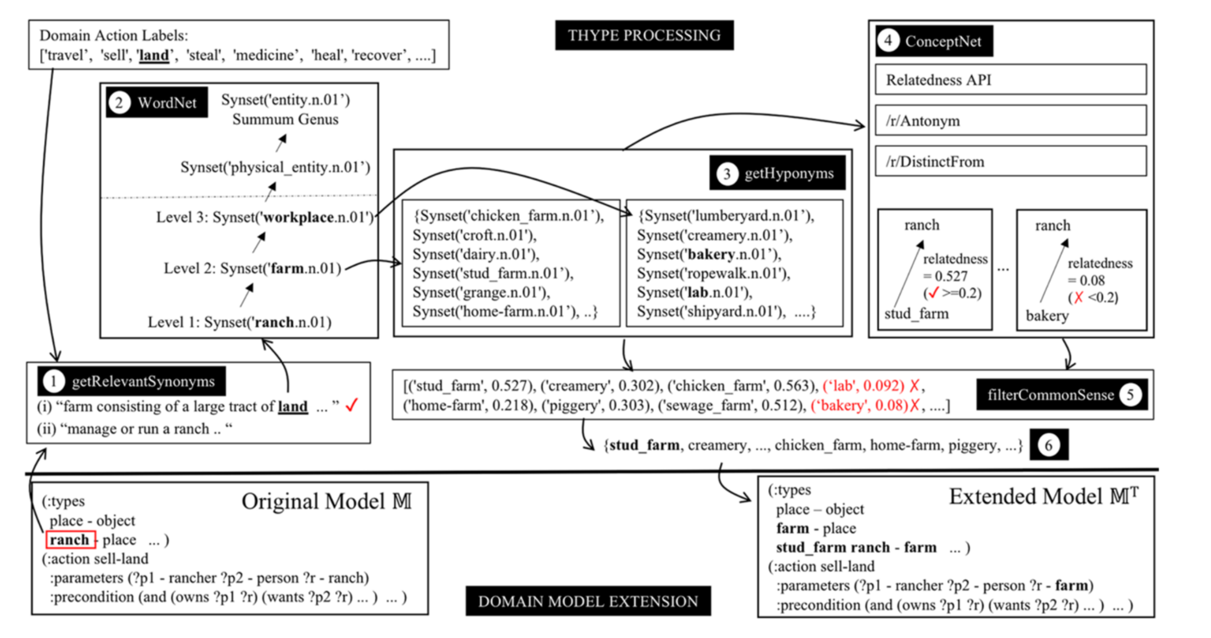
```
:action become-beautiful
  :parameters (?p - princess)
  :precondition (and (not (beautiful ?p)))
  :effect (and (beautiful ?p)))
```

```
(:action become-ugly
  :parameters (?p - princess)
  :precondition (and (beautiful ?p))
  :effect (and (not (beautiful ?p))))
```

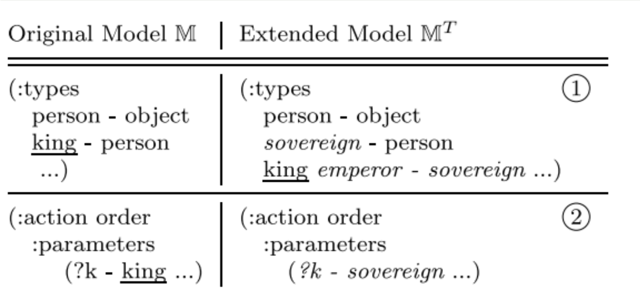
③ New Predicates: e.g. replace (not (beautiful ?p) with (ugly ?p)

```
(:action become-beautiful
  :parameters (?p - princess)
  :precondition (and (ugly ?p))
  :effect (and (beautiful ?p) (not (ugly ?p))))
```

- New actions are generated using the original action(marry) as a template for the newly generated action (divorce)
- The pre-conditions are those predicates achieved by the action A (green circled) plus an enabling condition; the positive effects are the pre-conditions of the initial action B (green circled)


Domain Action Labels:
['travel', 'sell', 'land', 'steal', 'medicine', 'heal', 'recover', ...]
THYPE PROCESSING
2 WordNet
Synset('entity.n.01') Summum Genus
Synset('physical_entity.n.01')
Level 3: Synset('workplace.n.01')
Level 2: Synset('farm.n.01)
Level 1: Synset('ranch.n.01)
3 getHyponyms
{Synset('chicken_farm.n.01'), Synset('croft.n.01'), Synset('dairy.n.01'), Synset('stud_farm.n.01'), Synset('grange.n.01'), Synset('home-farm.n.01'), ..}
{Synset('lumberyard.n.01'), Synset('creamery.n.01'), Synset('bakery.n.01'), Synset('ropewalk.n.01'), Synset('lab.n.01'), Synset('shipyard.n.01'), ...}
4 ConceptNet
Relatedness API
/r/Antonym
/r/DistinctFrom
ranch
relatedness = 0.527
(✓ >=0.2)
stud_farm
...
ranch
relatedness = 0.08
(✗ <0.2)
bakery
1 getRelevantSynonyms
(i) "farm consisting of a large tract of land ... " ✓
(ii) "manage or run a ranch .. "
[('stud_farm', 0.527), ('creamery', 0.302), ('chicken_farm', 0.563), ('lab', 0.092) ✗, ('home-farm', 0.218), ('piggery', 0.303), ('sewage_farm', 0.512), ('bakery', 0.08) ✗, ...]
filterCommonSense 5
{stud_farm, creamery, ..., chicken_farm, home-farm, piggery, ...} 6
Original Model 𝕄
(:types
place - object
ranch - place ...)
(:action sell-land
:parameters (?p1 - rancher ?p2 - person ?r - ranch)
:precondition (and (owns ?p1 ?r) (wants ?p2 ?r) ...) ...)
DOMAIN MODEL EXTENSION
Extended Model 𝕄ᵀ
(:types
place – object
farm - place
stud_farm ranch - farm ...)
(:action sell-land
:parameters (?p1 - rancher ?p2 - person ?r - farm)
:precondition (and (owns ?p1 ?r) (wants ?p2 ?r) ...) ...)

- takes an existing type from the original story domain (like ranch) and finds related types (like stud_farm, creamery, chicken_farm) to increase narrative diversity.



Original Model 𝕄	Extended Model $\mathbb{M}^T$
(:types person - object king - person ...)	(:types ① person - object *sovereign* - person king *emperor* - *sovereign* ...)
(:action order :parameters (?k - king ...)	(:action order ② :parameters (*?k* - *sovereign* ...)

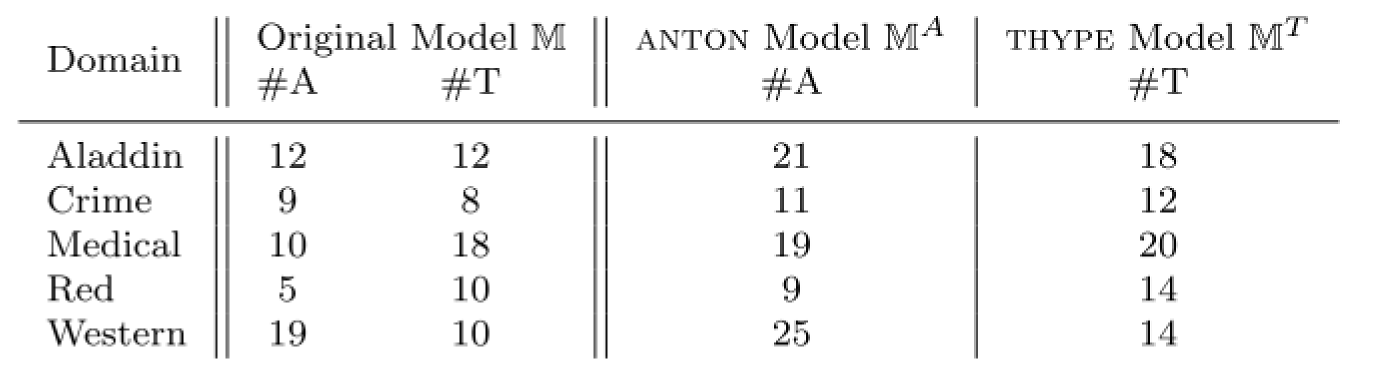
THYPE IN ACTION

Why does this matter?

- Diversity: More possible storylines leads to better replayability
- Robustness: If a user breaks the story, planner can still reach an ending
- Human-readability: All actions/types are labeled using real language

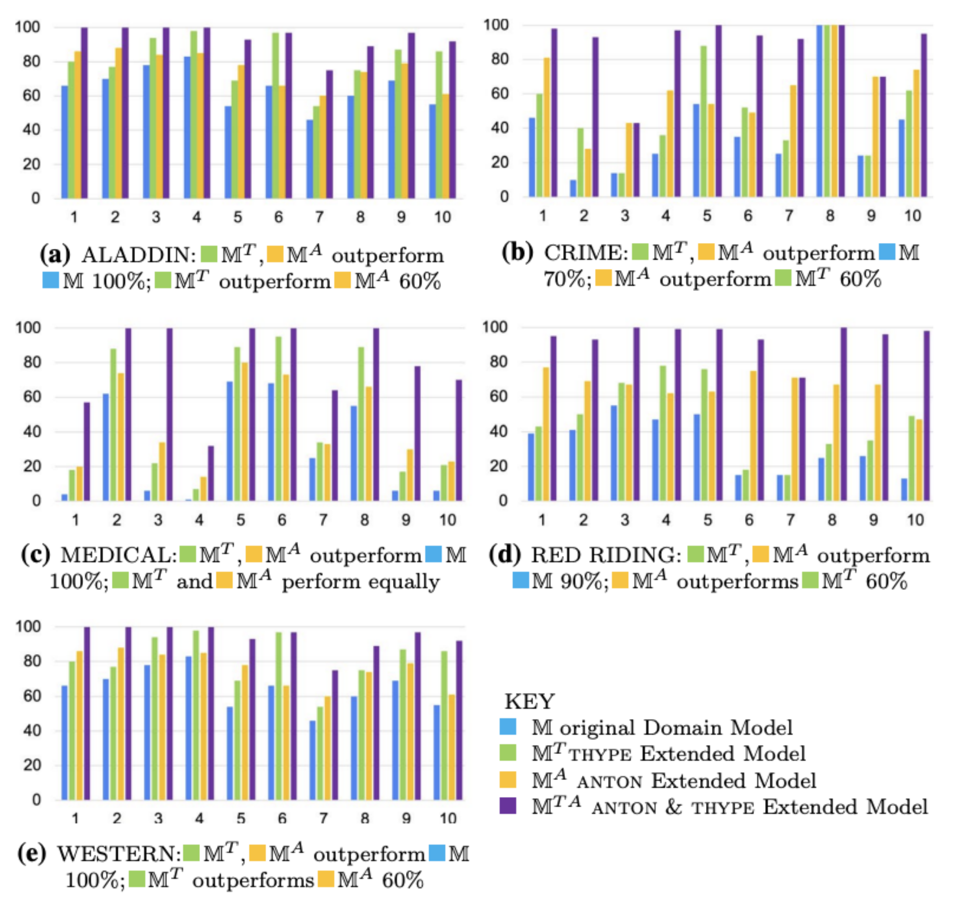
Evaluation

- Tested on a range of narrative planning domains: Aladdin, Crime Drama, Medical Drama, Red Riding Hood, Western
- M, M^T, M^A, M^{TA}



Domain	Original Model $\mathbb{M}$		ANTON Model $\mathbb{M}^A$	THYPE Model $\mathbb{M}^T$
	#A	#T	#A	#T
Aladdin	12	12	21	18
Crime	9	8	11	12
Medical	10	18	19	20
Red	5	10	9	14
Western	19	10	25	14

- We can observe that new actions and types were generated across all domains



(a) ALADDIN: 🟩$\mathbb{M}^T$, 🟨$\mathbb{M}^A$ outperform 🟦$\mathbb{M}$ 100%; 🟩$\mathbb{M}^T$ outperform 🟨$\mathbb{M}^A$ 60%

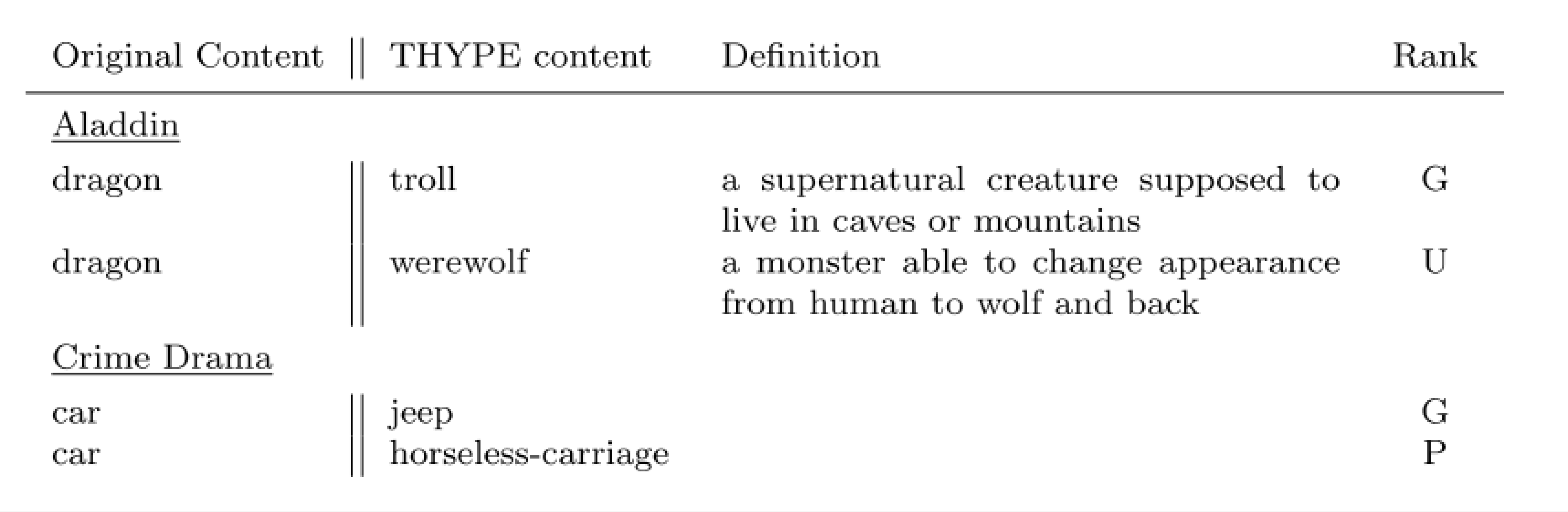
(b) CRIME: 🟩$\mathbb{M}^T$, 🟨$\mathbb{M}^A$ outperform 🟦$\mathbb{M}$ 70%; 🟨$\mathbb{M}^A$ outperform 🟩$\mathbb{M}^T$ 60%

(c) MEDICAL: 🟩$\mathbb{M}^T$, 🟨$\mathbb{M}^A$ outperform 🟦$\mathbb{M}$ 100%; 🟩$\mathbb{M}^T$ and 🟨$\mathbb{M}^A$ perform equally

(d) RED RIDING: 🟩$\mathbb{M}^T$, 🟨$\mathbb{M}^A$ outperform 🟦$\mathbb{M}$ 90%; 🟨$\mathbb{M}^A$ outperforms 🟩$\mathbb{M}^T$ 60%

(e) WESTERN: 🟩$\mathbb{M}^T$, 🟨$\mathbb{M}^A$ outperform 🟦$\mathbb{M}$ 100%; 🟩$\mathbb{M}^T$ outperforms 🟨$\mathbb{M}^A$ 60%

- Robustness Results: comparing ability of (green square)T, (yellow square) A, (purple square) TA to continue to original goal in execution failure simulation (5 domains, 10 problems, 100 fail-restarts on each)
- M(skyblue square) is included for comparison.
- Performance of T and A are consistently good across all domains and outperform throughout.
- TA outperforms all other models and demonstrates the performance gains that can result from combining extensions



Original Content	THYPE content	Definition	Rank
Aladdin			
dragon	troll	a supernatural creature supposed to live in caves or mountains	G
dragon	werewolf	a monster able to change appearance from human to wolf and back	U
Crime Drama			
car	jeep		G
car	horseless-carriage		P

- Examples of thype generated content and user rankings (G(ood), P(oor) and U(nsure)).
- For the type “dragon” (Aladdin), the alternative type, “troll” was ranked G, whereas the alternative, “werewolf”, was ranked U
- for “car” (Crime Drama) the type “jeep” was ranked G, and “horseless-carriage” P.
- These rankings are genre dependent and can differ across domains e.g. a werewolf is plausible in Harry Potter but not Aladdin.

Strengths

- Automates a major bottleneck in narrative AI development
- Generalizable to many story genres
- Human-readable output makes adoption easier
- Demonstrates measurable gains in robustness & diversity

Weaknesses

- Relies heavily on external linguistic tools and dictionaries which could lead to error
- Some generated actions/types are not appropriate for narrative
  - Requires human filtering (not 100% automatic)

Conclusion

- This paper presents an automated way to enhance narrative AI planning models
- It significantly improves diversity and robustness

Thank You!