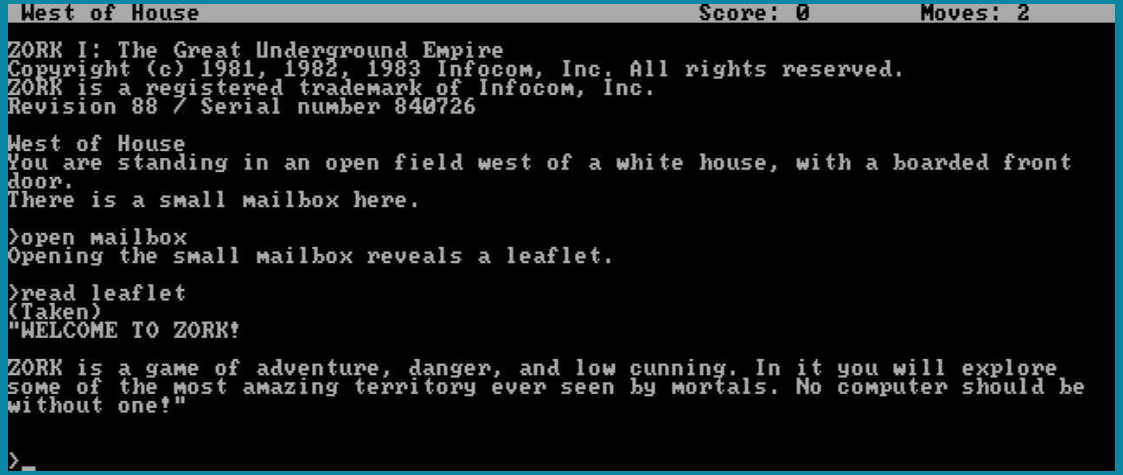
INTERACTIVE FICTION GAMES: A Colossal Adventure

Arnaud le Doeuff
Ignacio Dorado
Jorge Orbegozo

“ ¿WHAT ARE INTERACTIVE FICTION (IF) GAMES?

Fully text-based simulation environments where a player issues text commands to effect change in the environment and progress through the story



GOAL: test the efficiency of autonomous Reinforcement Learning agents playing IF games.

CHALLENGES:

Combinatorial Action Space

Decision making

+

Natural Language Processing

(700^4 = 240 billions)

Commonsense Reasoning

Open the chest?

or

Eat the chest?

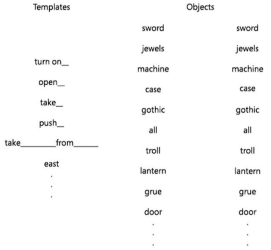
Knowledge Representation

Travel through the world

Keeping track of objects

Jericho

- Open source
- Infocom games
- More accessible for existing agents
- Point-based scoring sistem



Algorithm 1 Procedure for Identifying Valid Actions

$\mathcal{E} \leftarrow$ Jericho environment
$\mathcal{T} \leftarrow$ Set of action templates
$o \leftarrow$ Textual observation
$\mathcal{P} \leftarrow \{p_1 \dots p_n\}$ Interactive objects identified with noun-phrase extraction or world object tree.
$Y \leftarrow \varnothing$ List of valid actions
$s \leftarrow \mathcal{E}.save()$ – Save current game state
for template $u \in \mathcal{T}$ **do**
  for all combinations $p_1, p_2 \in \mathcal{P}$ **do**
    Action $a \leftarrow u \Leftarrow p_1, p_2$
    if $\mathcal{E}.world\_changed(\mathcal{E}.step(a))$ **then**
      $Y \leftarrow Y \cup a$
      $\mathcal{E}.load(s)$ – Restore saved game state
return Y

Handicaps:

1. Fixed random seed
2. Load, Save
3. Game templates and vocabulary
4. World object tree representation
5. World change detection

Algorithms

DRRN:

- Choice-based
- Single-game
- Based valid actions
- Update sampling minibatch
- Uses Jericho's handicaps

TDQN:

- Parser-based
- Single-game
- Templete based
- Supervised binary cross entropy loss
- Uses Jericho's handicaps

NAIL:

- Parser-based
- General-game
- Manually heuristic
- Web-based to decide
- No handicaps

Experiments

Agents were **evaluated** across a set of 32 games

Game	$\lvert\mathcal{T}\rvert$	$\lvert\mathcal{V}\rvert$	RAND	NAIL	TDQN	DRRN	MaxScore
905	82	296	0	0	0	0	1
acorncourt	151	343	0	0	1.6	**10**	30
advent†	189	786	36	36	36	36	350
adventureland	156	398	0	0	0	**20.6**	100
afflicted	146	762	0	0	1.3	**2.6**	75
anchor	260	2257	0	0	0	0	100
awaken	159	505	0	0	0	0	50
balances	156	452	0	**10**	4.8	**10**	51
deephome	173	760	1	**13.3**	1	1	300
detective	197	344	113.7	136.9	169	**197.8**	360

Results

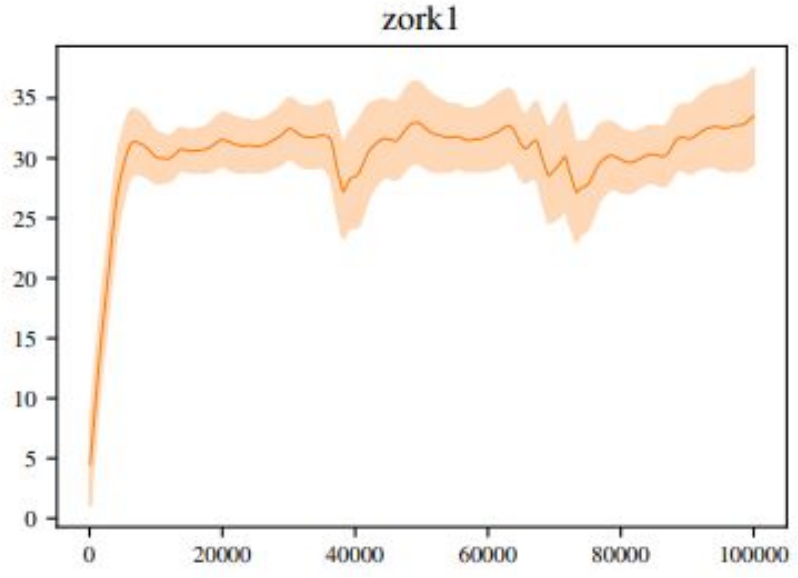
Agent	RAND	NAIL	TDQN	DRRN
Completion rate (%)	1.8	4.9	6.1	10.7

Tree difficulty tiers:

- Possible games
- Difficult games
- Extreme games

Example with Zork1

- 5 runs of each algorithm



Agents result for Zork 1

Game	|T |	|V|	RAND	NAIL	TDQN	DRRN	Max Score
Zork 1	237	697	0	10.3	9.9	32.6	350

Conclusions

- The fact that DRRN, the choice-based agent, outperformed TQDN, shows the **difficulty of language generation**.
- DRRN and TDQN were trained and evaluated on individual games, still far from a truly **general-purpose agent**
- TDQN algorithm computes independent Q-values for words and templates, **conditional generation** is an improvement yet to be explored.

QUESTIONS?