

ІІТМО

Граф знаний по игре Dota 2

Немного про игру...



- В игре **125 персонажей**
- Играется **5 на 5, 10 уникальных персонажей** в одной игре
- Каждый персонаж имеет свои характеристики и роли, а также преимущества и недостатки (неописанные, следующие из его характеристик и способностей) относительно любого другого героя
- Важной частью игры, особенно на высоком уровне, является выбор персонажа

Задача - составление графа знаний о героях и предметах игры Dota 2



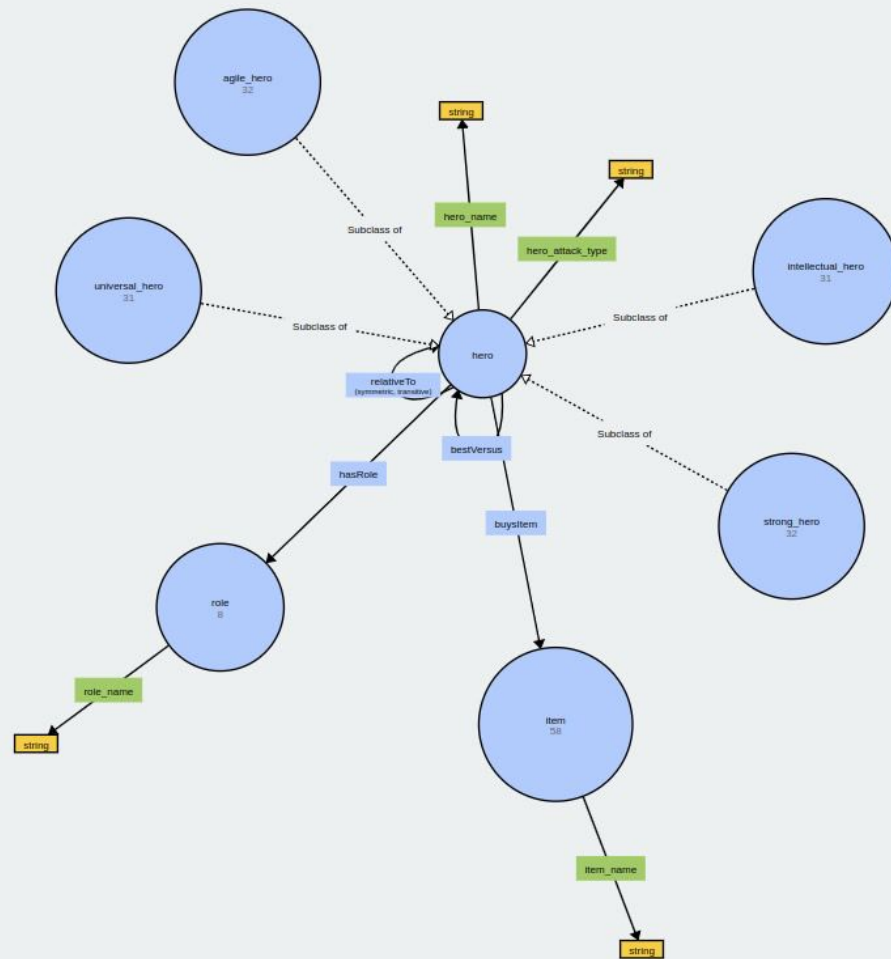
Пользователи - начинающие игроки. Граф решает следующие задачи:

- 1. Помощь в выборе героя, отталкиваясь от выбора героев противниками**
- 2. Помощь в выборе предмета для героя**
- 3. Предоставление некоторой интересной информации о истории героев**

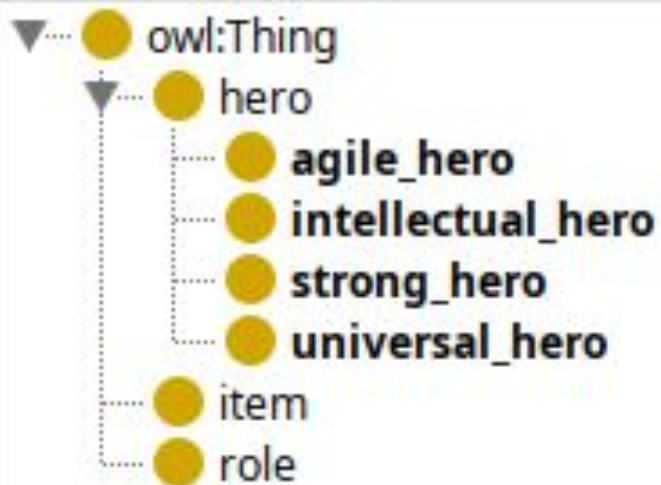


1. **OpenDota API** - информация о предметах и статистике выбора героев
2. **dota2.fandom.com** - информация об истории персонажей
3. **немного домыслов с Reddit**

Онтология



Структура классов



Object properties



A screenshot of a software interface, likely a web browser or a specialized application, displaying a tree view of object properties. The tree is rooted at 'owl:topObjectProperty' and contains four child items: 'bestVersus', 'buysItem', 'hasRole', and 'relativeTo'. Each item is preceded by a blue square icon. The interface has a white background with a grey border and a small minus and close button in the top right corner.

- owl:topObjectProperty
 - bestVersus
 - buysItem
 - hasRole
 - relativeTo

Герои, которые хорошо себя показывают в игре против своих родственников

```
query_1 = """
PREFIX dota2: <http://www.semanticweb.org/root/ontologies/2025/0/dota2#>
SELECT ?hero ?relative
WHERE {
  ?hero dota2:bestVersus ?relative .
  ?hero dota2:relativeTo ?relative .
}
"""
```



```
Герой: Dragon-Knight, хорош против родственника: Jakiro
Герой: Storm-Spirit, хорош против родственника: Void-Spirit
Герой: Zeus, хорош против родственника: Mars
```


Герои с наименьшим количеством ролей

```
query_3 = """
PREFIX dota2: <http://www.semanticweb.org/root/ontologies/2025/0/dota2#>
SELECT ?hero
WHERE {
  ?hero dota2:hasRole ?role .
}
GROUP BY ?hero
HAVING (COUNT(?role) < 3)
"""

for row in g.query(query_3):
    print(f"Герой с 1 или 2 ролями: {row.hero.replace('http://www.semanticweb.org/root/ontologies/2025/0/dota2#', '')}")
```

```
Герой с 1 или 2 ролями: Phantom-Assassin
Герой с 1 или 2 ролями: Weaver
Герой с 1 или 2 ролями: Dawnbreaker
Герой с 1 или 2 ролями: Shadow-Fiend
Герой с 1 или 2 ролями: Zeus
Герой с 1 или 2 ролями: Sniper
Герой с 1 или 2 ролями: Templar-Assassin
Герой с 1 или 2 ролями: Pugna
Герой с 1 или 2 ролями: Chen
Герой с 1 или 2 ролями: Lich
Герой с 1 или 2 ролями: Techies
Герой с 1 или 2 ролями: Bounty-Hunter
```

3 пары героев, которые хороши против одних и тех же героев

```
query_most_common_best_versus = """
PREFIX dota2: <http://www.semanticweb.org/root/ontologies/2025/0/dota2#>
SELECT ?hero1 ?hero2 (COUNT(?commonEnemy) AS ?commonCount)
WHERE {
  ?hero1 dota2:bestVersus ?commonEnemy .
  ?hero2 dota2:bestVersus ?commonEnemy .
  FILTER (?hero1 != ?hero2)
}
GROUP BY ?hero1 ?hero2
ORDER BY DESC(?commonCount)
LIMIT 3
"""
```



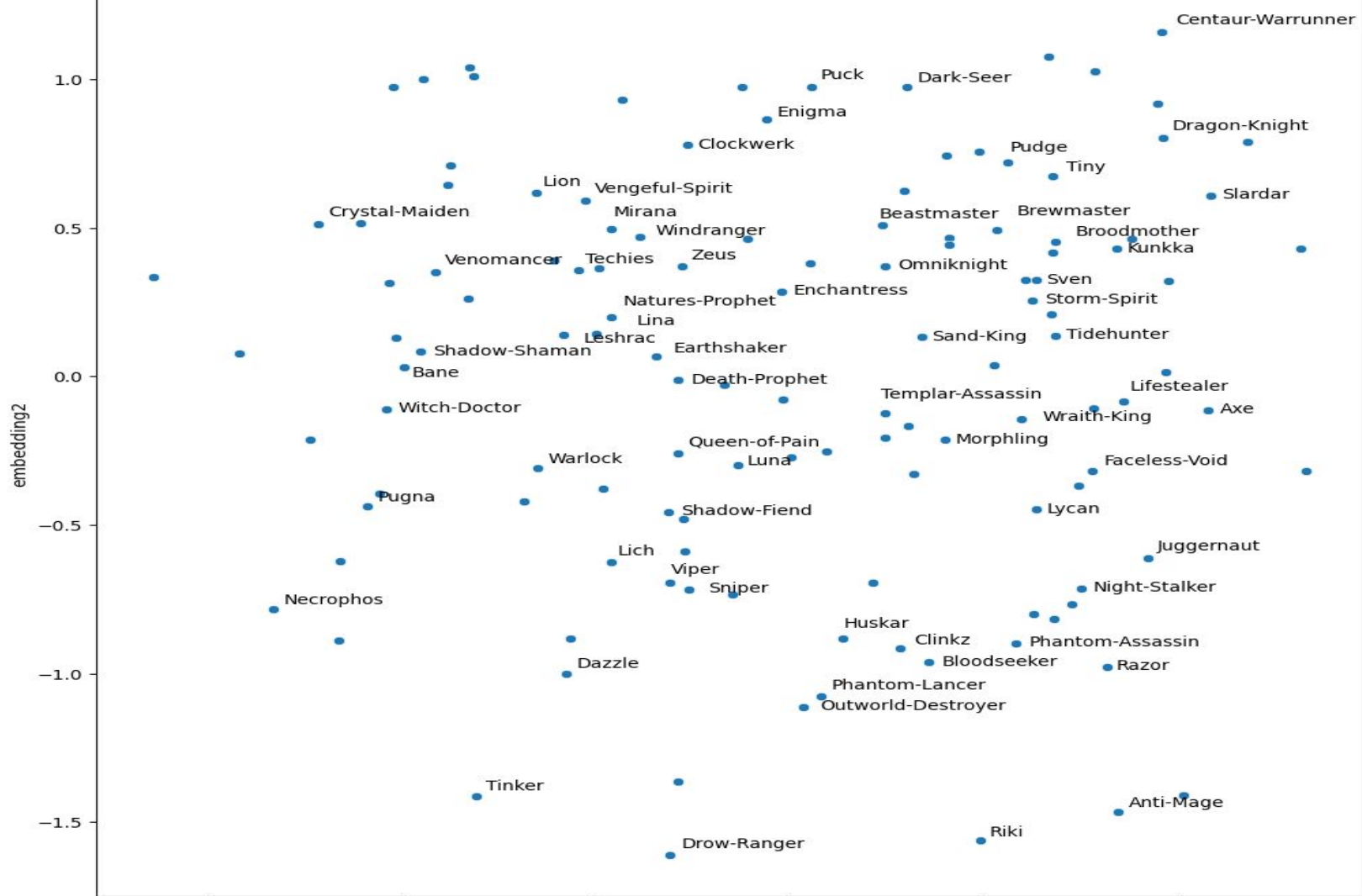
```
Герой 1: Alchemist, Герой 2: Arc-Warden, Общее количество bestVersus: 6
Герой 1: Zeus, Герой 2: Centaur-Warrunner, Общее количество bestVersus: 6
Герой 1: Centaur-Warrunner, Герой 2: Zeus, Общее количество bestVersus: 6
```

Найти наиболее подходящих героев против выбора соперника

ИТМО

```
query_5 = """
PREFIX dota2: <http://www.semanticweb.org/root/ontologies/2025/0/dota2#>
SELECT ?hero
WHERE {
  | ?hero dota2:bestVersus dota2:Dazzle .
  | ?hero dota2:bestVersus dota2:Snapfire .
}
"""
```

Герой, который хорош против Dazzle и Snapfire: Juggernaut
Герой, который хорош против Dazzle и Snapfire: Luna



MO



Ranks

```
mr = mr_score(ranks)
mrr = mrr_score(ranks)

print("MRR: %.2f" % (mrr))
print("MR: %.2f" % (mr))

hits_10 = hits_at_n_score(ranks, n=10)
print("Hits@10: %.2f" % (hits_10))
hits_3 = hits_at_n_score(ranks, n=3)
print("Hits@3: %.2f" % (hits_3))
hits_1 = hits_at_n_score(ranks, n=1)
print("Hits@1: %.2f" % (hits_1))
```

```
MRR: 0.28
MR: 34.26
Hits@10: 0.40
Hits@3: 0.30
Hits@1: 0.20
```

Link Prediction

```
from scipy.special import expit
probs = expit(scores)

pd.DataFrame(list(zip([' '.join(x) for x in statements],
                      ranks,
                      np.squeeze(scores),
                      np.squeeze(probs))),
             columns=['statement', 'rank', 'score', 'prob']).sort_values("prob")
```

	statement	rank	score	prob
2	Pudge bestVersus Night-Stalker	[89, 84]	0.988217	0.728736
3	Pudge bestVersus Oracle	[1, 6]	4.950305	0.992968
0	Pudge bestVersus Axe	[1, 6]	5.062058	0.993707
1	Pudge bestVersus Tinker	[1, 2]	5.598040	0.996309

**Спасибо
за внимание!**

it'sMO *re than a*
UNIVERSITY