

MASWS – Report 3

Razvan Ranca – 0954584

I've written several queries that answer interesting questions about the poker dataset from a data exploration perspective. For each query I've included a "description" of what the query is trying to accomplish and the reasons for wanting to accomplish this, and an "observations" section talking about what insights we might draw from looking at the query results.

The queries make use of all the information encoded in the dataset, especially the temporal ordering of events and the statistical information regarding the likely winners and the strength of the hands held by the players.

The queries can be run on the full dataset, but execution takes a few minutes. For convenience I ran them on a restricted dataset 10% the size of the full one. This smaller dataset still has 300 played hands, so it should be sufficient for our purposes. The dataset is called "smallMatch.n3" and can be found on https://github.com/RazvanRanca/RDF_Poker

To save space, I'll omit the prefixes from the query texts. So all queries can be assumed to have the following prefixes:

PREFIX rdfs: <<http://www.w3.org/2000/01/rdf-schema#>>

PREFIX rdf: <<http://www.w3.org/1999/02/22-rdf-syntax-ns#>>

PREFIX poker_terms: <http://dbpedia.org/resource/Glossary_of_poker_terms#>

PREFIX poker_schema: <<http://homepages.inf.ed.ac.uk/s0954584/poker-schema.xml/#>>

PREFIX xsd: <<http://www.w3.org/2001/XMLSchema#>>

All queries shown below can also be found at:

https://github.com/RazvanRanca/RDF_Poker/tree/master/queries

Example queries:

Query 1

Description: One of the easiest signals to pick up about a player's style is how often he "goes all-in" (i.e. makes a bet where he put all his money in). This first query recovers all hands where someone went "all-in"

Query:

```
SELECT ?hh ?pn
```

```
WHERE
```

{ ?d rdfs:type poker_schema:Deal .	#all deals
?d poker_terms:stack ?stack .	#stack size for deal
?d rdf:_1 ?fbr .	#first betting round
?fbr poker_schema:before_betting_round* ?br .	#all betting rounds
?br poker_schema:before_bet* ?bet .	#all bets in all betting rounds
?bet poker_terms:act ?act .	#the act corresponding to each bet
?act poker_schema:action_type "r" .	#restrict to acts which are raises
?act rdf:value ?stack .	#restrict to raises of the full stack (i.e. "all-in" raises)
?act poker_schema:acting_player ?p .	#the player who went all in
?p poker_schema:played_by ?pn .	#the name of this player
?d rdfs:comment ?hh . }	#the hand history, for display purposes
LIMIT 10	# limit to 10 results

Result:

```
| "STATE:185:cr200c/r300r450c/cr20000f:Qd7c|4cJc/9d2dJh/2c:-450|450:dcubot|azure_sky" |
<http://www.computerpokercompetition.org/downloads/competitions/2012/logs/#2012-2p-
nolimit/azure_sky> |
| "STATE:89:cc/r200c/r479r20000f:TdKc|2sQc/3cQs7s/5c:-479|479:dcubot|azure_sky" |
<http://www.computerpokercompetition.org/downloads/competitions/2012/logs/#2012-2p-
nolimit/azure_sky> |
| "STATE:187:cc/cr200c/r450r20000f:6sQh|KhAd/9s5s9h/9d:-450|450:dcubot|azure_sky" |
<http://www.computerpokercompetition.org/downloads/competitions/2012/logs/#2012-2p-
nolimit/azure_sky> |
| "STATE:90:cc/r200c/r20000f:QsTc|6s8s/6cAsQc/3s:200|-200:azure_sky|dcubot" |
<http://www.computerpokercompetition.org/downloads/competitions/2012/logs/#2012-2p-
nolimit/azure_sky> |
| "STATE:152:r200c/r300c/r20000f:JsTd|JcTc/Jd5s4s/9h:300|-300:azure_sky|dcubot" |
<http://www.computerpokercompetition.org/downloads/competitions/2012/logs/#2012-2p-
nolimit/azure_sky> |
| "STATE:189:cc/cc/cr20000f:Ks8d|JsKh/6s5cKd/Qd:-100|100:dcubot|azure_sky" |
<http://www.computerpokercompetition.org/downloads/competitions/2012/logs/#2012-2p-
nolimit/azure_sky> |
| "STATE:252:r200c/r300c/r20000f:8cQs|3d6d/3cQdTc/Ac:300|-300:azure_sky|dcubot" |
<http://www.computerpokercompetition.org/downloads/competitions/2012/logs/#2012-2p-
nolimit/azure_sky> |
| "STATE:76:r228c/r342c/r20000f:9cTc|4cAh/Ts3c4s/Ac:342|-342:azure_sky|dcubot" |
<http://www.computerpokercompetition.org/downloads/competitions/2012/logs/#2012-2p-
nolimit/azure_sky> |
| "STATE:239:cc/cr200c/r500r20000f:5s8c|As4c/5h8s8d/5d:-500|500:dcubot|azure_sky" |
<http://www.computerpokercompetition.org/downloads/competitions/2012/logs/#2012-2p-
nolimit/azure_sky> |
| "STATE:108:r256c/cc/r20000f:KhKc|Qd9d/Ac7s3s/As:256|-256:azure_sky|dcubot" |
<http://www.computerpokercompetition.org/downloads/competitions/2012/logs/#2012-2p-
nolimit/azure_sky> |
```

Observation: There are several things to note here, but the most obvious is that every single all-in bet is from the player “azure_sky”. It seems this player is much more aggressive than his opponent.

Query2

Description: Since “azure_sky” seems to like to bet aggressively it would be interesting to see if he also tries to intimidate his opponent by bluffing. A classic bluffing scenario is when 3 community cards of the same suit come on the flop and one player bets a lot even though his cards do not match. Essentially he is trying to send the message that he has cards of the same suit and thus a very strong hand, even though in reality he does not. We can test for this situation.

Query:

```
SELECT ?pn
```

```
WHERE
```

```
{ ?br poker_schema:betting_stage "Flop" .           #all betting rounds at the "Flop" stage
  ?br poker_schema:before_bet* ?bet .               #all bets in betting rounds on the "Flop"
  ?bet poker_schema:bets_cur_round 0 .              #restrict to bets that were first in their betting round
  ?bet <http://dbpedia.org/resource/Pot_(poker)> ?p . #the current pot amount
  ?bet poker_terms:act ?act .                       #the betting action taken
  ?act rdf:value ?betVal .                          #the amount bet
  ?act poker_schema:acting_player ?bp .             #the player making the bet
  ?bp poker_schema:played_by ?pn .                  #that player's name
```

```

?bp rdf:_1 ?pc1 . #the betting player's cards
?bp rdf:_2 ?pc2 .
?pc1 poker_schema:card_suit ?ps1 . #the suit of the betting player's cards
?pc2 poker_schema:card_suit ?ps2 .
?bet poker_terms:community_card ?cc . #the available community cards
?cc rdf:_1 ?cc0 .
?cc rdf:_2 ?cc1 .
?cc rdf:_3 ?cc2 .
?cc0 poker_schema:card_suit ?cs0 . #the suits of the community cards
?cc1 poker_schema:card_suit ?cs0 .
?cc2 poker_schema:card_suit ?cs0 .
FILTER(!(sameTerm(?ps1,?cs0) || sameTerm(?ps2,?cs0))) #retain situations where the player
cards suits were different from the community cards suits(i.e the player has a weak hand)
FILTER(?betVal > ?p) #retain situations where the player bet a large
amount(more than the previous pot). In conjunction with the above this means he's bluffing
}
LIMIT 10 # limit to 10 results

```

Result:

```

| <http://www.computerpokercompetition.org/downloads/competitions/2012/logs/#2012-2p-
nolimit/dcubot> |
| <http://www.computerpokercompetition.org/downloads/competitions/2012/logs/#2012-2p-
nolimit/dcubot> |
| <http://www.computerpokercompetition.org/downloads/competitions/2012/logs/#2012-2p-
nolimit/dcubot> |
| <http://www.computerpokercompetition.org/downloads/competitions/2012/logs/#2012-2p-
nolimit/dcubot> |
| <http://www.computerpokercompetition.org/downloads/competitions/2012/logs/#2012-2p-
nolimit/dcubot>

```

Observation: This is a very interesting result. All bluffs belong to “dcubot”. So it seems that “azure_sky” likes to bet large sums, but probably only when he actually has a good hand. “dcubot” on the other hand seems to enjoy bluffing.

Query 3:

Description: After the previous results it's still unclear who the more aggressive player is. Another classic statistic to use is the amount of times a player has reached the “show-down”. This occurs when neither player folds in any of the betting rounds, so they have to compare hands and the strongest wins. Not folding is a sign of aggressiveness, so this should be a good measure.

Query:

```

SELECT ?pn (COUNT(?act) as ?noActs)
WHERE
{ ?brs poker_schema:betting_stage "River" . #all betting rounds at the final stage (i.e the "River")
  ?brs poker_schema:before_bet+ ?bet . #all bets after the initial one
  OPTIONAL {?bet poker_schema:before_bet ?nextBet} . #hold only the bets that are last in the
chain(i.e. they have no "nextBet")
  FILTER ( !BOUND(?nextBet) ) .
  ?bet poker_terms:act ?act . #act performed on the final bet
  ?act poker_schema:action_type ?actType . #type of act
  FILTER (?actType != "f") . #retain only those acts which weren't folds
  ?act poker_schema:acting_player ?bp . #the player making the act

```

```
?bp poker_schema:played_by ?pn .           #the player's name
}
GROUP BY ?pn                                #group by the player's name
```

Result:

|<http://www.computerpokercompetition.org/downloads/competitions/2012/logs/#2012-2p-nolimit/azure_sky>|8|

|<<http://www.computerpokercompetition.org/downloads/competitions/2012/logs/#2012-2p-nolimit/dcubot>> | 45

Observation: Well, this settles it. “dcubot” is clearly more aggressive, having reached the showdown 45 times as opposed to “azure_sky”’s 8

Query 4

Description: Now that we know some basic facts about the player's aggressiveness we should investigate their play style more closely. Usually an informative thing to see is how the players handle either them or the opponent having a stroke of good luck at the end of the game. To investigate this we must make use of the winning probabilities that were introduced in the data via pre-processing.

Query:

```
SELECT ?hh ?wn
```

WHERE

```
{ ?d rdf:type poker_schema:Deal .                #all deals
  ?d poker_schema:player1 ?f .                    #the first player
  ?d poker_schema:player2 ?s .                    #the second player
  ?d rdf:_1 ?fbr .                                #the first betting round
  ?fbr poker_schema:before_betting_round* ?br .   #all betting rounds
  ?br poker_schema:betting_stage "River" .        #restrict to betting rounds at the "River" stage
  ?prevBr poker_schema:before_betting_round ?br . #the previous betting round
```

```
{ ?d poker_schema:winner ?f .           #first subset, restrict to the winner = first player
  ?br poker_schema:win_prob_P1 ?p1 .     #the probability p1 has of winning
  ?br poker_schema:win_prob_P2 ?p2 .     #the probability p2 has of winning
  ?prevBr poker_schema:win_prob_P1 ?prevP1 . #the prob. p1 had of winning in the prev. round
  ?prevBr poker_schema:win_prob_P2 ?prevP2 . #the prob. p2 had of winning in the prev. round
  FILTER (?p1 > ?p2) .                  #keep cases where p1 is winning in the current round
  FILTER (?prevP1 < ?prevP2) }          #but was losing in the previous round
```

UNION

```
{ ?d poker_schema:winner ?s . #second subset, same as above only for p2 winning instead of p1
?br poker_schema:win_prob_P1 ?p1 .
?br poker_schema:win_prob_P2 ?p2 .
?prevBr poker_schema:win_prob_P1 ?prevP1 .
?prevBr poker_schema:win_prob_P2 ?prevP2 .
FILTER (?p2 > ?p1) .
FILTER (?prevP2 < ?prevP1)}
```

```

?d poker_schema:winner ?w .           #the winner
?w poker_schema:played_by ?wn .        #the winner's name
?d rdfs:comment ?hh .                  #the hand history
}

```

LIMIT 10 #limit to 10 results

Results:

"STATE:167:cc/cr200c/r300r600r900r1800c/r1900r2850f:6h9c|3cKc/3d5dTs/6d/Kh:-1900|1900:dcubot|azure_sky" |
<http://www.computerpokercompetition.org/downloads/competitions/2012/logs/#2012-2p-nolimit/azure_sky> |
| "STATE:243:cc/r200r400c/r638r1276c/r1376c:AdTd|2cAh/6s3s5d/5h/4h:-1376|1376:dcubot|azure_sky" |
<http://www.computerpokercompetition.org/downloads/competitions/2012/logs/#2012-2p-nolimit/azure_sky> |
| "STATE:146:r207r724c/r1086c/r1629c/r2443r3257r4885c:8d6h|9d7s/8h7c5c/6s/8s:4885|-4885:azure_sky|dcubot" |
<http://www.computerpokercompetition.org/downloads/competitions/2012/logs/#2012-2p-nolimit/azure_sky> |
| "STATE:75:cc/r485c/r585r877c/r1590r2385r3180r6360c:JsQs|ThJc/Td4sAs/3h/Kc:6360|-6360:dcubot|azure_sky" |
<http://www.computerpokercompetition.org/downloads/competitions/2012/logs/#2012-2p-nolimit/dcubot> |
| "STATE:212:r422c/r844c/r1012c/r1518c:5dAs|3dJh/2hTc4s/Js/3c:1518|-1518:azure_sky|dcubot" |
<http://www.computerpokercompetition.org/downloads/competitions/2012/logs/#2012-2p-nolimit/azure_sky> |
| "STATE:12:r200r500c/r1000c/r1500c/r2250c:3d6d|QsAd/4h6c8c/8s/Qd:-2250|2250:azure_sky|dcubot" |
<http://www.computerpokercompetition.org/downloads/competitions/2012/logs/#2012-2p-nolimit/dcubot> |
| "STATE:36:r394c/r591c/r886r1181r1771r2361r3541c/r5311c:Jd9d|6hAh/Jh7h5c/Ad/Jc:5311|-5311:azure_sky|dcubot" |
<http://www.computerpokercompetition.org/downloads/competitions/2012/logs/#2012-2p-nolimit/azure_sky> |
| "STATE:49:cc/cc/r200r300c/r995c:8c4c|AdAh/Kh3c8s/Ts/4h:995|-995:dcubot|azure_sky" |
<http://www.computerpokercompetition.org/downloads/competitions/2012/logs/#2012-2p-nolimit/dcubot> |
| "STATE:123:cc/r200r400c/r782r1564c/cr2346c:9d6h|Ks3s/6s2s7d/3d/Ts:-2346|2346:dcubot|azure_sky" |
<http://www.computerpokercompetition.org/downloads/competitions/2012/logs/#2012-2p-nolimit/azure_sky> |
| "STATE:136:r384c/cc/r484r1158r2316c/r3474c:8h9h|KcQs/Kd7h8d/Jd/9c:3474|-3474:azure_sky|dcubot" |
<http://www.computerpokercompetition.org/downloads/competitions/2012/logs/#2012-2p-nolimit/azure_sky> |

Observation: Analysing these hands can yield many observations. For instance, “dcubot” tends to call a lot, even if his opponent is showing signs of being strong. This is a weakness that can be exploited. On the other hand “azure_bot” isn’t very subtle. Every time he improved his hand he bet big. This confirms our earlier hunch from the all-in analysis that “azure_bot” tends to bet big whenever he has a good hand.

Query 5

Description: From the previous analysis it seems that neither player is afraid of playing for large sums. It would then be interesting to identify hands in which both players raised numerous times. Such a pattern is relatively rare and usually means that either both players are genuinely certain they have the better hand or are certain they can intimidate the other player into folding. Therefore displaying the hands with the most raises should give interesting information.

Query:

```
SELECT ?hh (COUNT (?act) as ?noRaises)
```

```
WHERE
```

```
{ ?d rdf:type poker_schema:Deal .           #all deals
  ?d rdfs:comment ?hh .                     #the hand history
  ?d rdf:_1 ?fbr .                          #first betting round
  ?fbr poker_schema:before_betting_round* ?br . #all betting rounds
  ?br poker_schema:before_bet* ?bets .       #all bets in all betting rounds
  ?bets poker_terms:act ?act .              #acts corresponding to all bets
  ?act poker_schema:action_type "r" .       #restrict to acts that are raises
}
```

```
GROUP BY ?hh
```

```
#group by the hand history, which is unique for every deal
```

```
HAVING (?noRaises > 7)
```

```
#limit to those with more than 7 raises
```

```
ORDER BY DESC(?noRaises)
```

```
#show raises in descending order
```

Results:

```
| "STATE:107:r200c/r328r492r656r984c/r1084r1626c/r1726r2589r3884r5826c:KdJh|
7s6d/Jd7d7h/Js/8d:5826|-5826:dcubot|azure_sky" | 11 |
| "STATE:162:r200r300c/r450c/r675r1057r1585r2300r3450c/r5175r6900r10350c:AhAs|
4cQh/Qc3d4s/Jh/6c:-10350|10350:azure_sky|dcubot" | 11 |
| "STATE:191:r200c/r300r600c/cr900r1200r1800c/r2129r3193c:6d2d|3s6s/6cJsTd/3d/Ah:-3193|
3193:dcubot|azure_sky" | 8 |
| "STATE:254:r200c/r400c/r500r1019r2038c/r3057r4076r6114f:5cQc|As3c/TdJcKh/Qd/Js:4076|-
4076:azure_sky|dcubot" | 8 |
| "STATE:36:r394c/r591c/r886r1181r1771r2361r3541c/r5311c:Jd9d|6hAh/Jh7h5c/Ad/Jc:5311|-
5311:azure_sky|dcubot" | 8 |
| "STATE:44:cc/cr200r300c/r450r734r1101c/r1651r2201r3301f:AhKh|5sAs/8h4h3h/Jc/Jh:2201|-
2201:azure_sky|dcubot" | 8
```

Observations: It's interesting to note that out of the top 6 hands regarding number of raises, dcubot has lost 4. This in conjunction with what we've seen earlier regarding the number of bluffs dcubot makes and his tendency to call even when he is at a disadvantage seems to indicate that dcubot is too aggressive. On the other hand the biggest sum out of these 6 hands, 10350 in the second hand on the list, was won by dcubot. We'd need more data to make a conclusion, but this may indicate that azure_sky's tendency to bet very large sums could be a liability for him.

Query 6

Description: Returning to the luck factor. We've seen how the players react when luck struck at the end of a game, but what about if one player is lucky from the start? Can the players translate a lucky start into a win? Investigating this is slightly more complex and requires two queries.

Query 6.1 Description: In this first query we use a CONSTRUCT statement to add triples showing the amount of luck involved at the start of the game (measured as the difference between relative strengths of hands between the two players on the first betting round). Note that "hand strength" is a stronger measure than the "probability to win" we used before. If both players hold a pair, then their hand strength is equal, but the probability to win will usually not be. Also note that the "luck" is considered from the point of view of player 1. So a luck of "+5" means player 1 is very lucky and a luck of "-5" means player 1 is very unlucky (and therefore player 2 is lucky)

Query 6.1:

[illegible]

Query 6.1 Results:

```
<http://www.computerpokercompetition.org/downloads/competitions/2012/logs/#2012-2p-
nolimit/match0_deal112>
    poker_schema:p1Lucky
    0 .
<http://www.computerpokercompetition.org/downloads/competitions/2012/logs/#2012-2p-
nolimit/match0_deal51>
    poker_schema:p1Lucky
    0 .
<http://www.computerpokercompetition.org/downloads/competitions/2012/logs/#2012-2p-
nolimit/match0_deal148>
    poker_schema:p1Lucky
    -2 .
<http://www.computerpokercompetition.org/downloads/competitions/2012/logs/#2012-2p-
nolimit/match0_deal44>
    poker_schema:p1Lucky
    5 .
etc.
```

Query 6.2 Description: The second query uses the usual dataset + a file containing the output of query 6.1. It displays the number of times player1 and player2 won grouped by the luck factor computed in 6.1

Query 6.2:

```

SELECT (SUM(?p1Won) AS ?p1Wins) (SUM(?p2Won) AS ?p2Wins) ?p1Luck
WHERE {
  SELECT (xsd:integer(?w=?p1) AS ?p1Won) (xsd:integer(?w=?p2) AS ?p2Won) ?p1Luck #p1Won
holds 1 if player1 won and 0 otherwise, p2Won does the same for player2
  WHERE { ?d rdf:type poker_schema:Deal .      #all deals
          ?d poker_schema:p1Lucky ?p1Luck .    #the previously computed "luck"
          ?d poker_schema:winner ?w .          #the winner
          ?d poker_schema:player1 ?p1 .        #the first player
          ?d poker_schema:player2 ?p2          #the second player
        } }
GROUP BY ?p1Luck                                #group by "luck"
ORDER BY DESC(?p1Luck)                          #order by descending "luck"

```

Query 6.2 results:

p1Wins	p2Wins	p1Luck	
=====			
1	1	5	
1	0	4	
0	1	3	
4	6	2	
35	21	1	
52	62	0	
14	41	-1	
3	4	-2	
0	1	-4	

Observations: The interesting observation here is that the most benefit of luck seems to come when the luck isn't huge. The lucky players seems to win more from the +/-1 cases, where the hands are comparable, and not from the ones with larger values. For instance, when having an advantage of 2 player 1 only manages to win in 4 out of 10 cases and player 2 manages to win 4 out of 7. This hints at a flaw in the players, namely that they do not know how to play very strong hands.

7. Description: Finally, since one of the main benefits of RDF is the merging of datasets, we want to test this capability by drawing some information from dbpedia. The problem is that our dataset only contains games between AI agents, however it should not be too difficult to generate a similar dataset for games between actual humans. As a proof-of-concept a fictitious match is generated between the professional players "Phil Hellmuth" and "Phil Ivey". This match is called "humanPlay.n3" and can also be found at: https://github.com/RazvanRanca/RDF_Poker Using this, we interact with dbpedia by retrieving the date of birth and best ranking in the World Series of Poker for both players.

Query:

PREFIX dbpedia-owl: <http://dbpedia.org/ontology/>

PREFIX dbpprop: <http://dbpedia.org/property/>

PREFIX foaf: <http://xmlns.com/foaf/0.1/>

PREFIX poker_schema: <http://homepages.inf.ed.ac.uk/s0954584/poker-schema.xml/#>

SELECT DISTINCT ?player ?birthDate ?wsopBestPlace

WHERE

```
{ ?p1 poker_schema:played_by ?sn .           #the surnames of both players
  SERVICE <http://dbpedia.org/sparql>        #use the dbpedia sparql endpoint
  { ?player foaf:surname ?sn .                #all people with same surname as the players
    ?player a dbpedia-owl:PokerPlayer .      #restrict to poker players
    ?player dbpedia-owl:birthDate ?birthDate . #birthdates of the players
    ?player dbpprop:wsopMainEventBestFinishRank ?wsopBestPlace . #their best ranking in the
World Series of Poker
  }
}
```


Results:

player	birthDate	wsopBestPlace
=====		
=====		
< http://dbpedia.org/resource/Phil_Hellmuth > "1964-07-16"^^< http://www.w3.org/2001/XMLSchema#date > "Winner"@en		
< http://dbpedia.org/resource/Phil_Ivey > "1976-02-01"^^< http://www.w3.org/2001/XMLSchema#date > 7		
=====		

Advantages of RDF:

As seen in query 7, using different datasets together, including local and remote ones, is quite easy. This means there is a potentially huge advantage to RDF over other querying methods in that, if more people start transforming poker related data into RDF, it will be easy to query all the datasets together. Even at the present, if one had a dataset of games between professional players, it would be possible to check things such as the ranking of poker players from dbpedia and see how this corresponds to their win percentage in your dataset.

Additionally, as seen in the first 6 queries, a large variety of questions regarding the data can be answered with relative ease via sparql. Each of those queries would have required a separate script to sort and process the data, scripts which would have usually taken longer to write and which would have been less flexible. For instance, with the above queries I can simply change the "GROUP BY" clause and go from seeing most re-raises in a deal to most re-raises in a betting round. Or from highest amount wagered in a match to highest amount wagered in a hand. All of these would require non-trivial changes to implement in a script.

I think this flexibility shows that RDF + sparql are excellent tools to use for purposes such as exploratory data analysis.