# Agenda

- What is microbenchmark?
- Why microbenchmarks for QA?
- QaldGenDataset for microbenchmarking
- QaldGen microbenchmarking framework

# What is Microbnechmark?

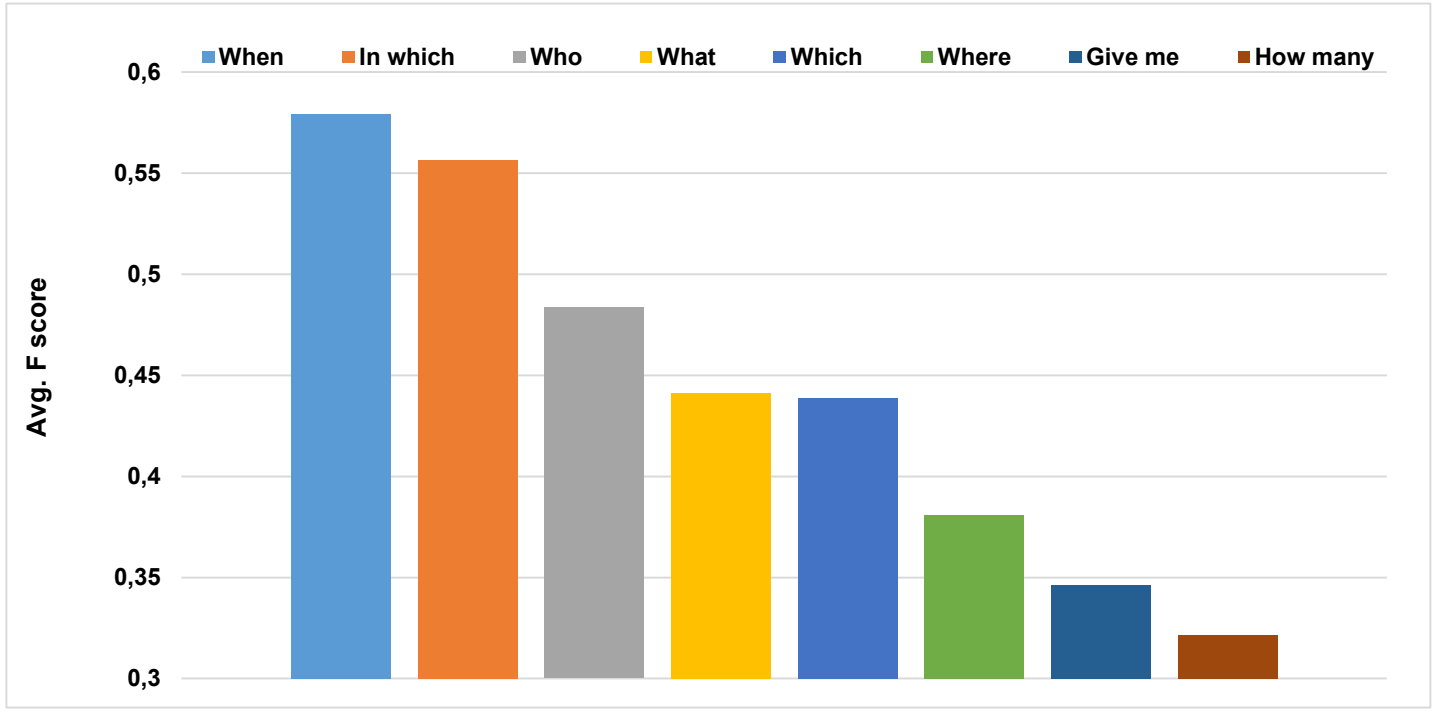
- Specialized
- Focused
- Use-case specific
- Useful for componenet-level testing
- E.g, single relation QA benchmark

# Why Microbnechmarks for QA?

- Multiple QA-related factors are influecing the F scores
  - Fine-grained testing of the systems
  - Pinpoint more detailed limitations
- Some systems are designed for special purpose or use-case
  - Avoids comparing apple with oranges

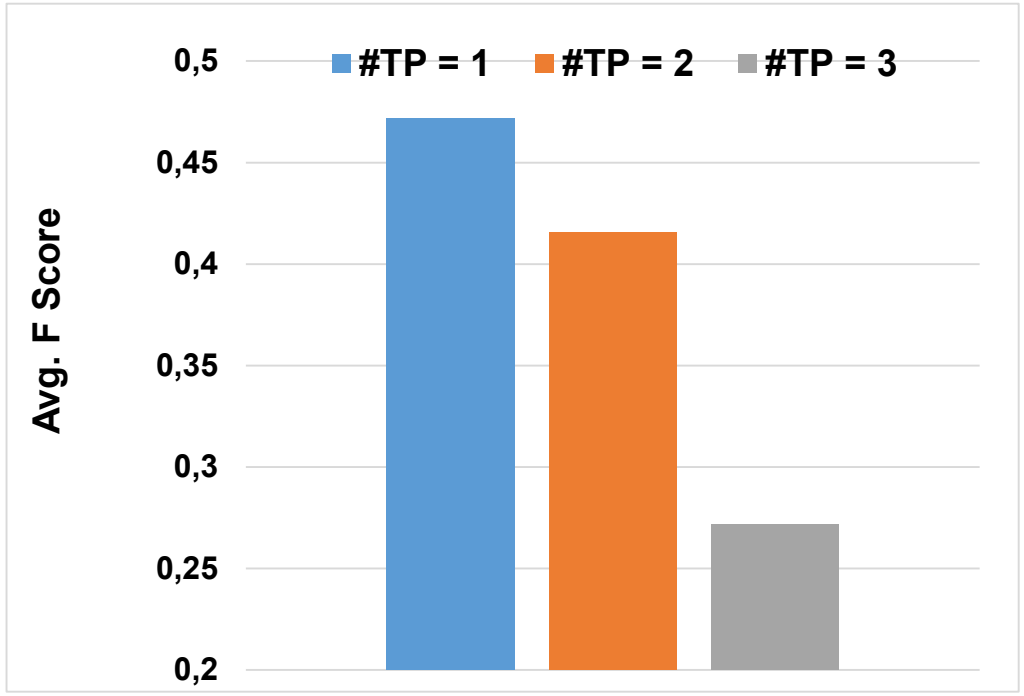
# Effect of Question Type

From QALD6 results



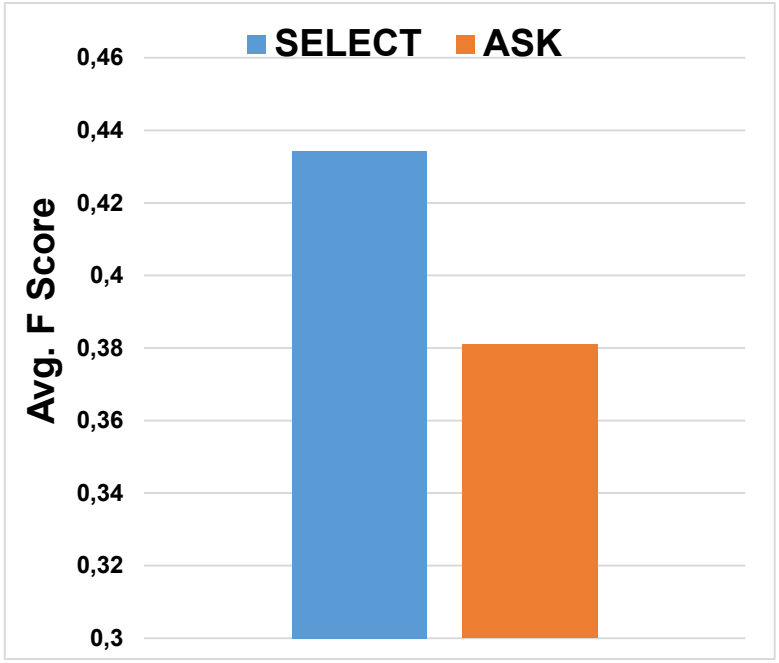

Easy → Difficult

# Effect of #Triple Patterns



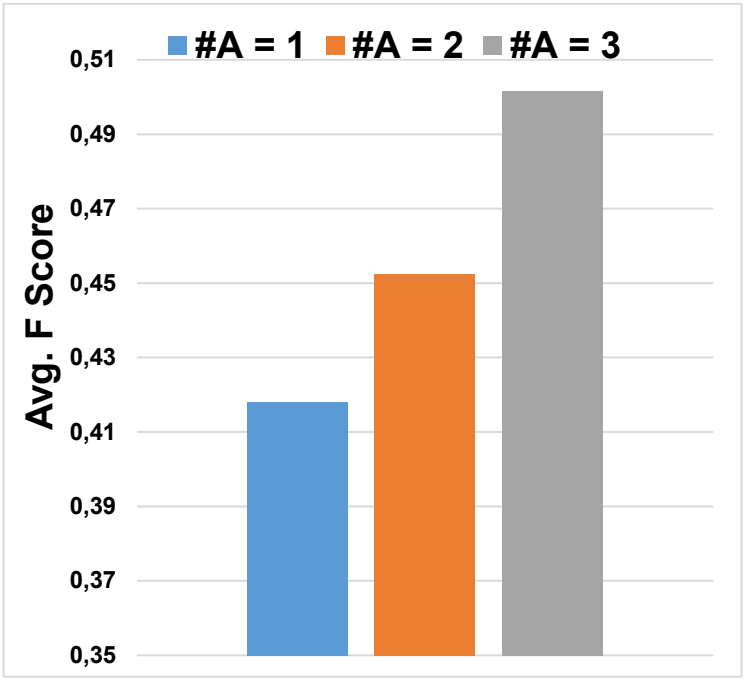

#Triple Patterns has inverse relationship with the F score, i.e., the more the Triple patterns the harder the question to answer

# Comparison on SPARQL query forms



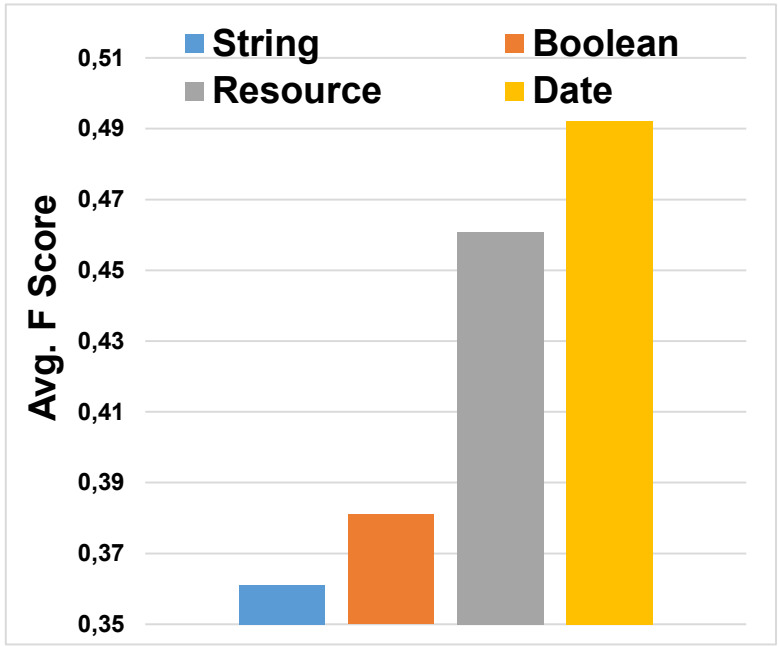

SPARQL ASK queries are much difficult to answer as compared to SPARQL SELECT

# Effect of #Answer



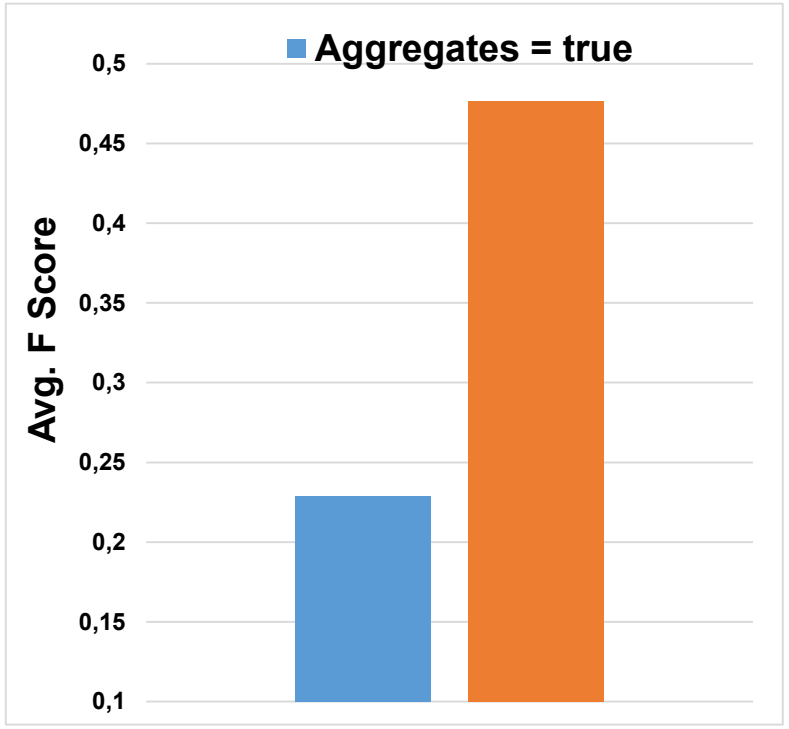

#Answers have direct relationship with the F score, i.e., the more answer the easiest the question to answer

# Effect of Answer Types




When answer is type of Date then it is more easy to answer and when answer is type of string it  is difficult to answer

# Effect of Aggregates functions




If the SPARQL query corresponding to a question has aggregate functions then it is difficult to answer

# Why Micobenchmarking for QA?

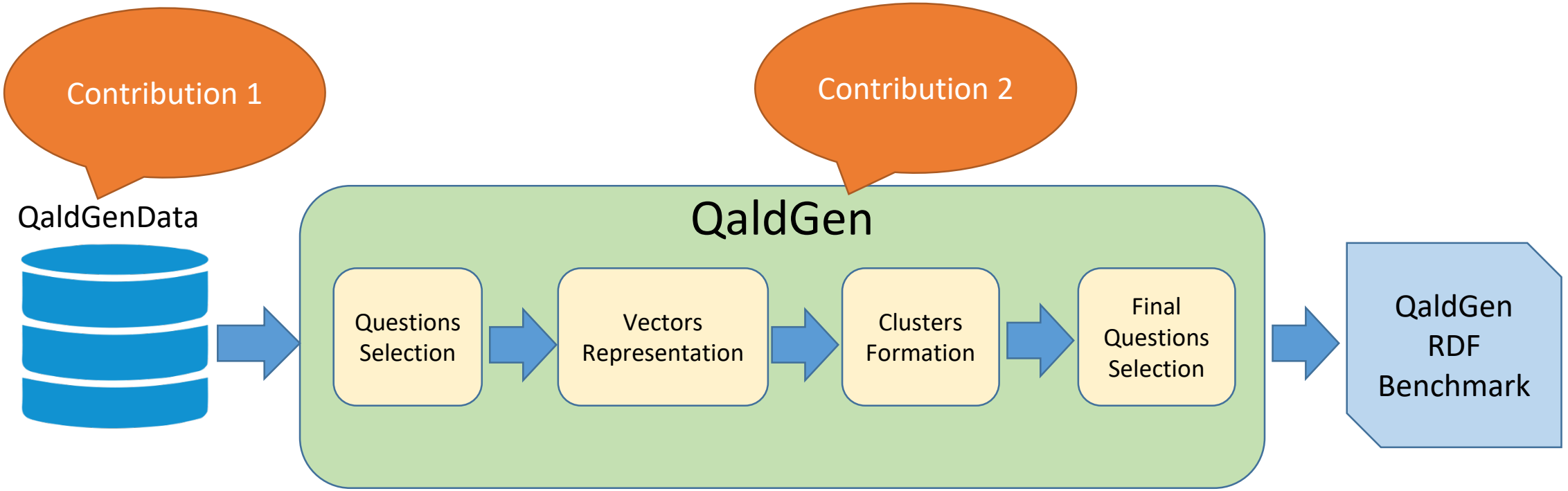
- QALD-6 Winner: CANALI QA System
  - ***But***, for questions starting "Give me" UTQA is winner
- NLIWOD SPARQL Query Builder has F-score 0.48 -was reported as baseline over LC-QuAD
  - ***But***, SINA on same dataset for two triples patterns in SPARQL queries has F-Score 0.80
  - ***But***, SINA reports F-score 0.0 for SPARQL queries with four triples


Definition of "baseline" is subjected to type of the questions, and specific features of the questions

# What is QaldGen?

- Micorbenchmarks selector framework
- For QA over Linked Data
- Personalized benchmarking
- Reusing LC-Quad and QALD
- Based on graph clustering

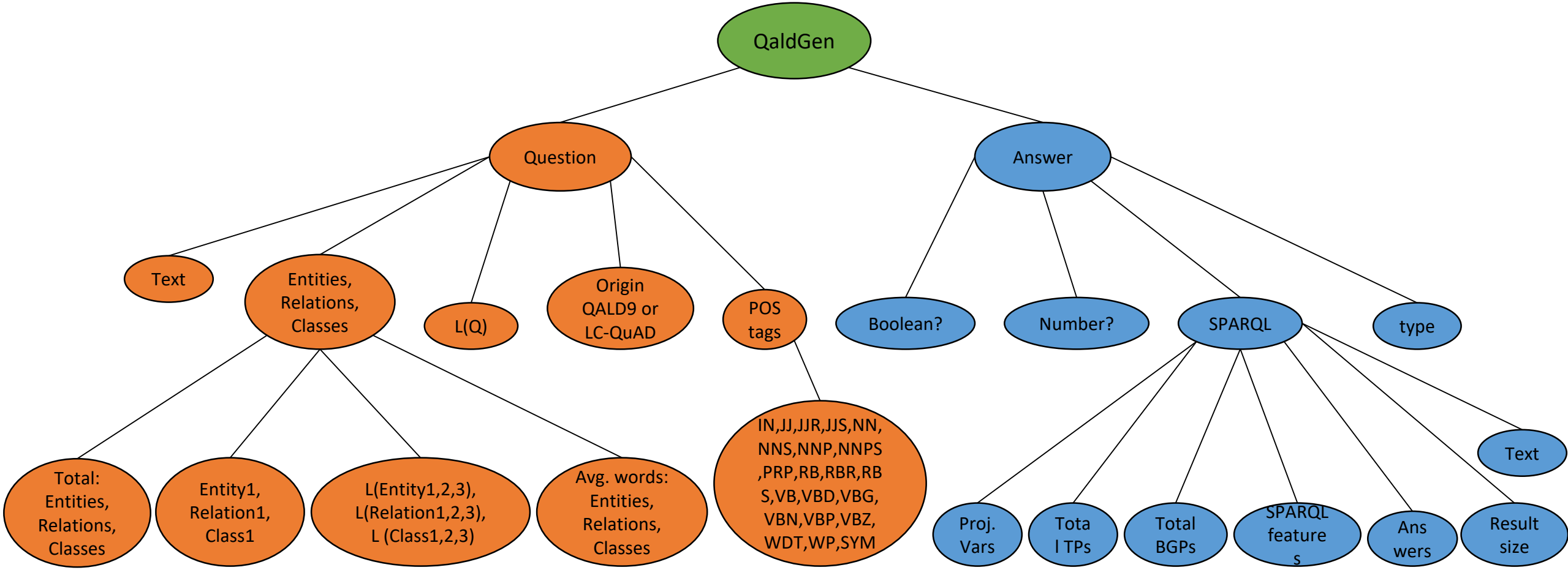
# QaldGen Architecture



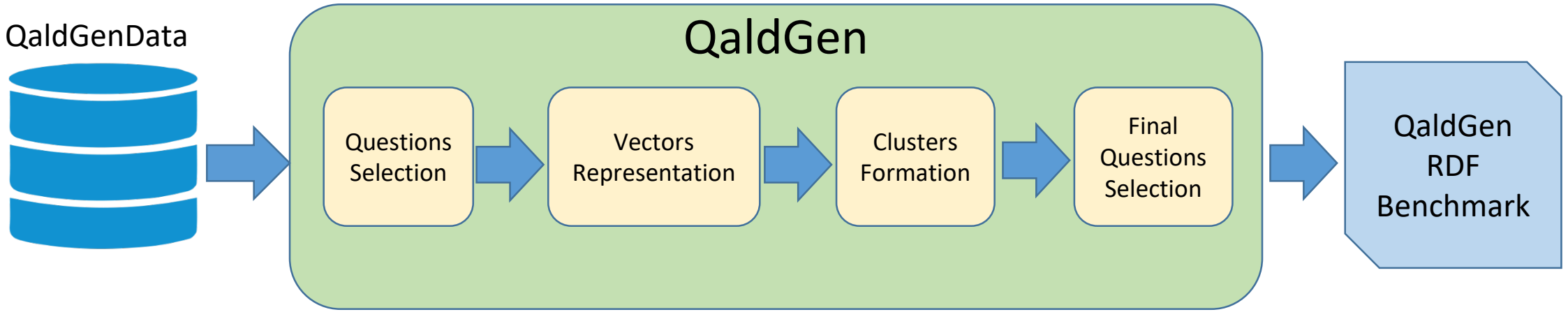
# QaldGenData

- RDF Dataset for QA over linked data
  - NER related testing
  - Relation linking testing
  - QA systems testing
- Constructed from LC-Quad and QALD-9
  - 408 questions from QALD-9
  - 5000 questions from LC-Quad
- Annotated each question with 51 features

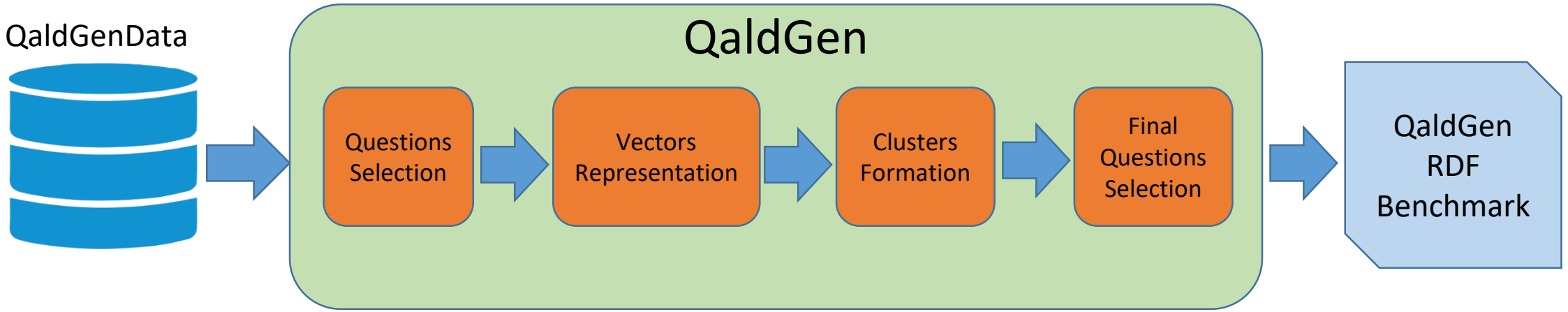
# QaldGenData: Annotations



# QaldGen: Micorobenchark Selection



# QaldGen: Micorobenchark Selection



# Step1: Questions Selection

- From QaldGenData
- Using a single SPARQL query
- Select desired features as projections
- Options for personalization
  - Use-case specific benchmarking
  - Using SPARQL constructs, e.g., Filters

```
Prefix qaldGen: <http://qald-gen.aksw.org/vocab#>
Prefix lsq: <http://lsq.aksw.org/vocab#>
SELECT  DISTINCT ?qId ?totalWords ?totalEntities ?
    totalRelations ?totalClasses ?avgEntitiesWords ?tps ?rs ?
    bgps ?pvars
{
?qId  qaldGen:length ?totalWords .
?qId  qaldGen:totalEntities ?totalEntities .
?qId  qaldGen:totalRelations ?totalRelations .
?qId  qaldGen:totalClasses ?totalClasses .
?qId  qaldGen:avgEntitiesWords ?avgEntitiesWords .
?qId  lsq:tps ?tps .
?qId  lsq:resultSize  ?rs .
?qId  lsq:bgps   ?bgps .
?qId  lsq:projectVars ?pvars .

# Options for Personalisation
?qId  qaldGen:questionOrigin "qald9" .
?qId  qaldGen:questionType  ?qType .
Filter Regex (?qType, "What")
Filter (?tps > 1 && ?rs >0)
}
```

# Step2: Vectors Representation

Annotated features as projections

Question Related:

- Total Words: 9
- Total Entities: 1
- Total Relations: 1
- Total Classes: 0
- Avg. Entities Words: 1

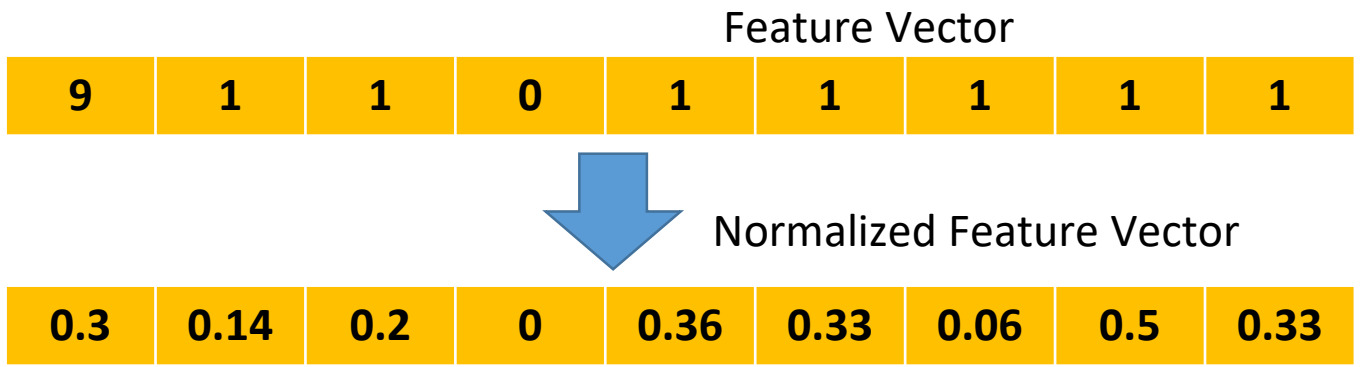
Answer Related

- #Triple patterns: 1
- #Results: 1
- #BGPs: 1
- #Projection Vars: 1

Question: " What is the time zone of SaltLake City?

Golden Answer:

```
SELECT DISTINCT ?uri
WHERE
 {
<http://dbpedia.org/resource/Salt_Lake_City>
<http://dbpedia.org/ontology/timeZone> ?uri
 }
```

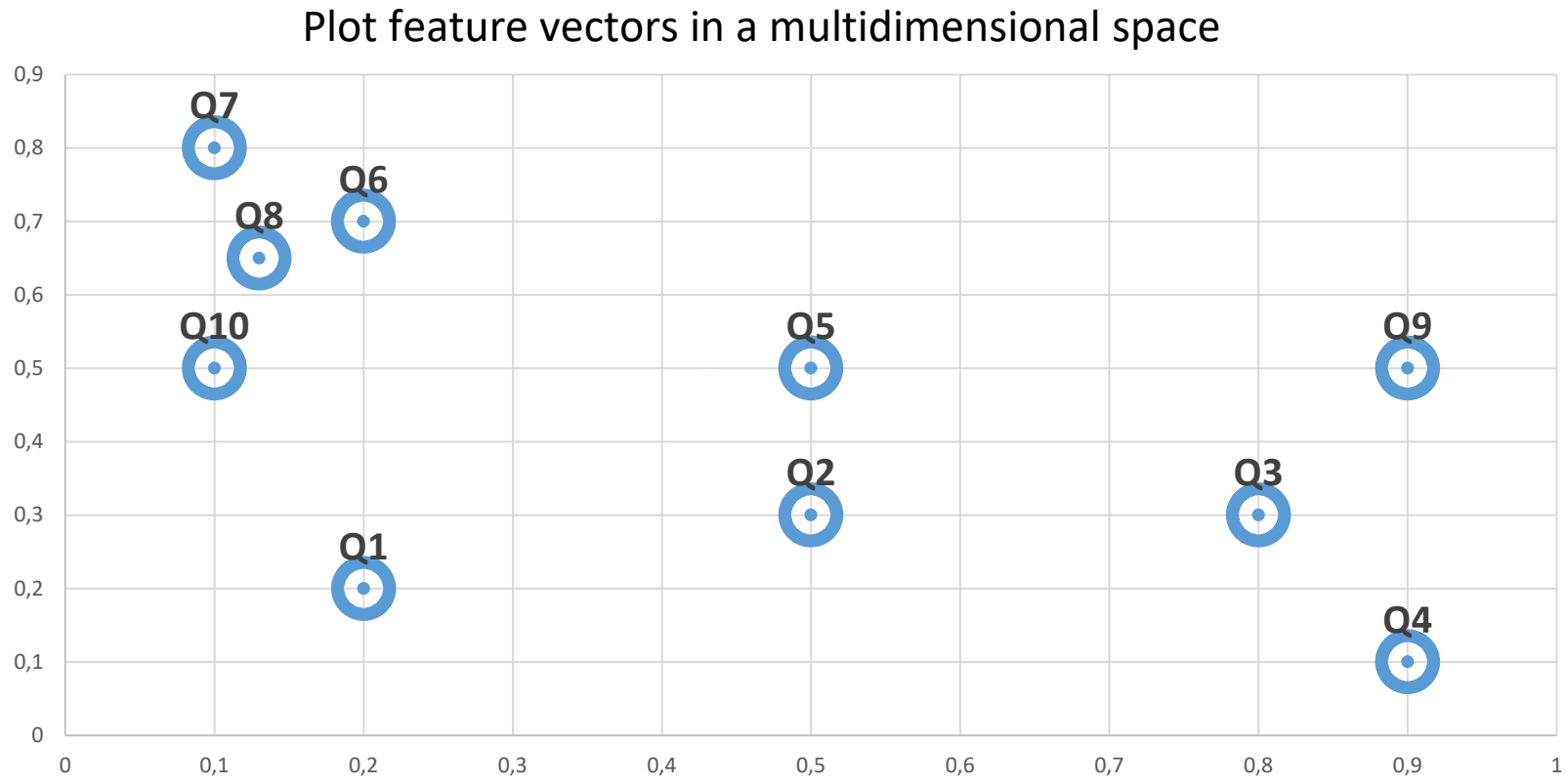


# Step3: Clusters Formation

- We do clustering to get sufficient diversity in the selected benchmark
- Draw vectors on multi-dementional space and get clusters
- QaldGen supports
  - Kmeans++
  - FEASIBLE
  - Agglomerative
  - Random
  - FEASIBLE-Exemplars
  - DBSCAN+Kmeans++
- Others clustering techniques can easily be integrated

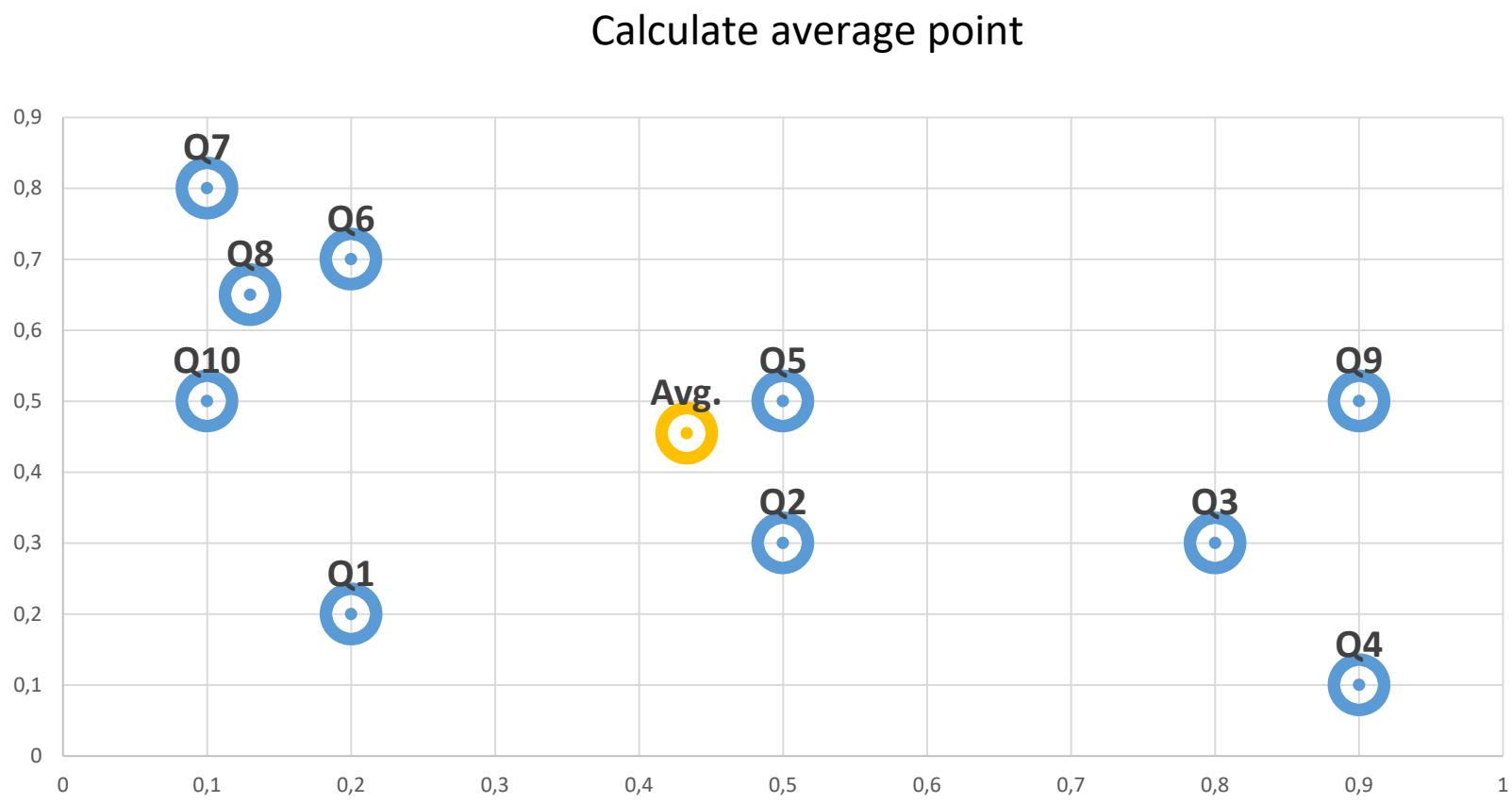
# Step3: Clusters Formation

| Q | F1 | F2 |
|---|---|---|
| Q1 | 0.2 | 0.2 |
| Q2 | 0.5 | 0.3 |
| Q3 | 0.8 | 0.3 |
| Q4 | 0.9 | 0.1 |
| Q5 | 0.5 | 0.5 |
| Q6 | 0.2 | 0.7 |
| Q7 | 0.1 | 0.8 |
| Q8 | 0.13 | 0.65 |
| Q9 | 0.9 | 0.5 |
| Q10 | 0.1 | 0.5 |



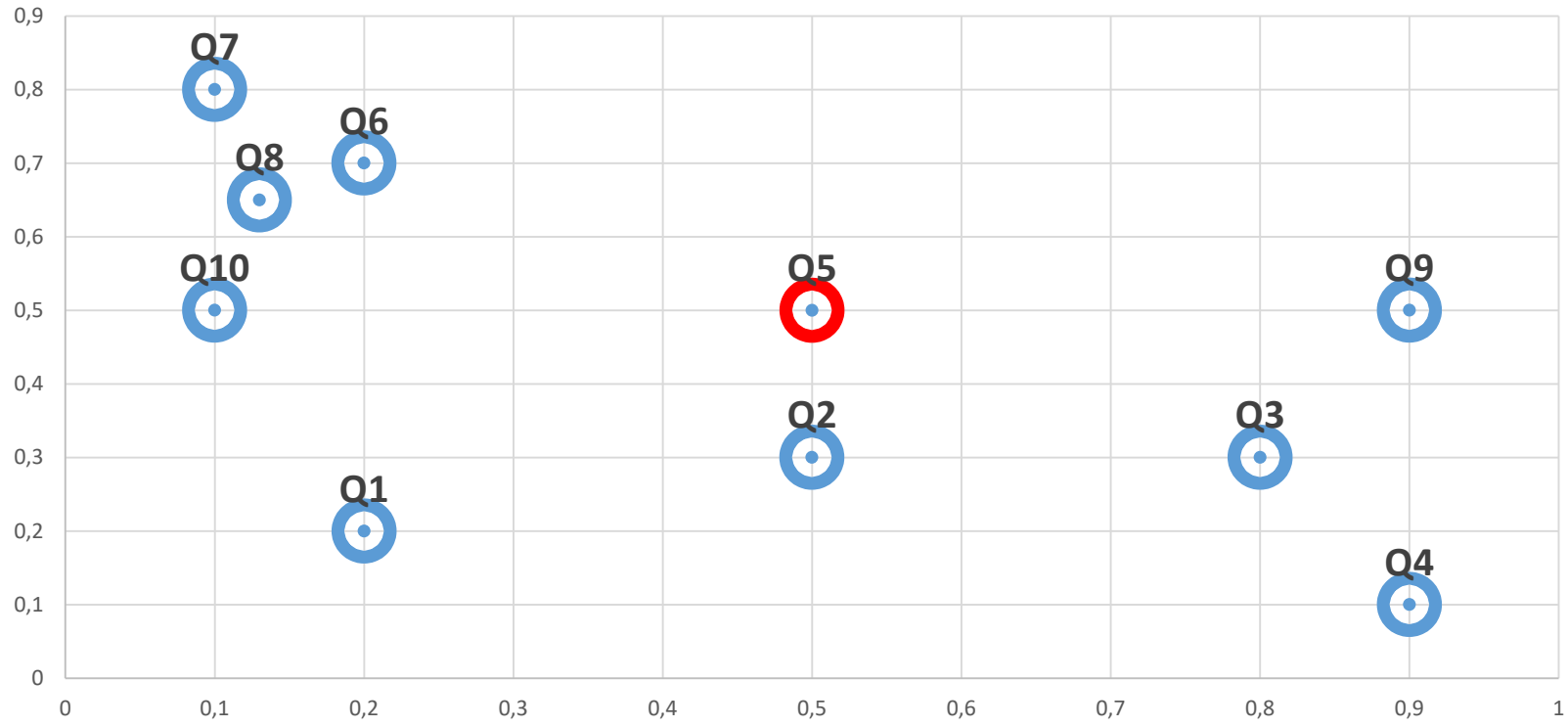

Suppose we need a benchmark of 3 questions

# Step3: Clusters Formation



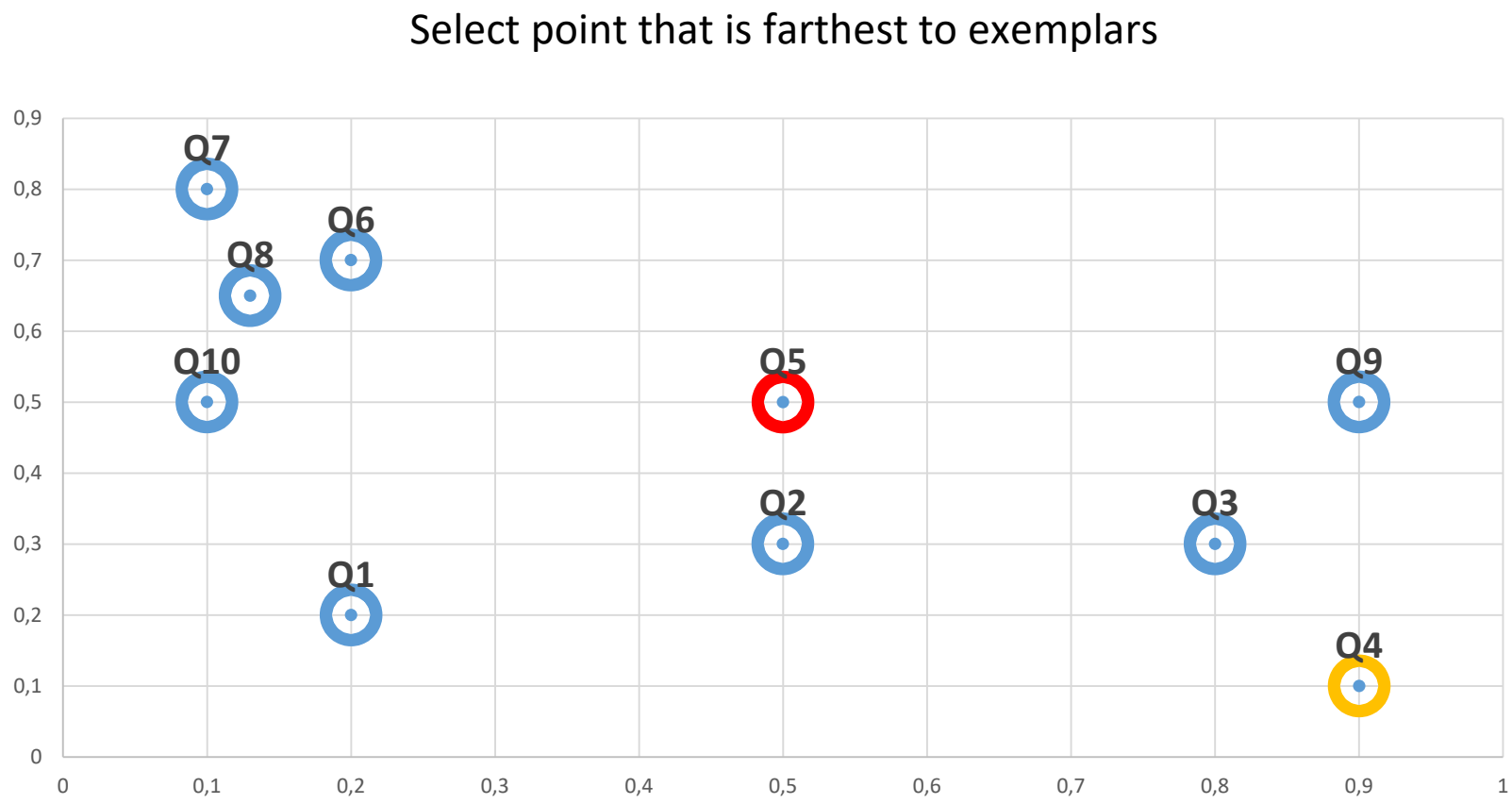
# Step3: Clusters Formation

Select point of minimum Euclidean distance to avg. point

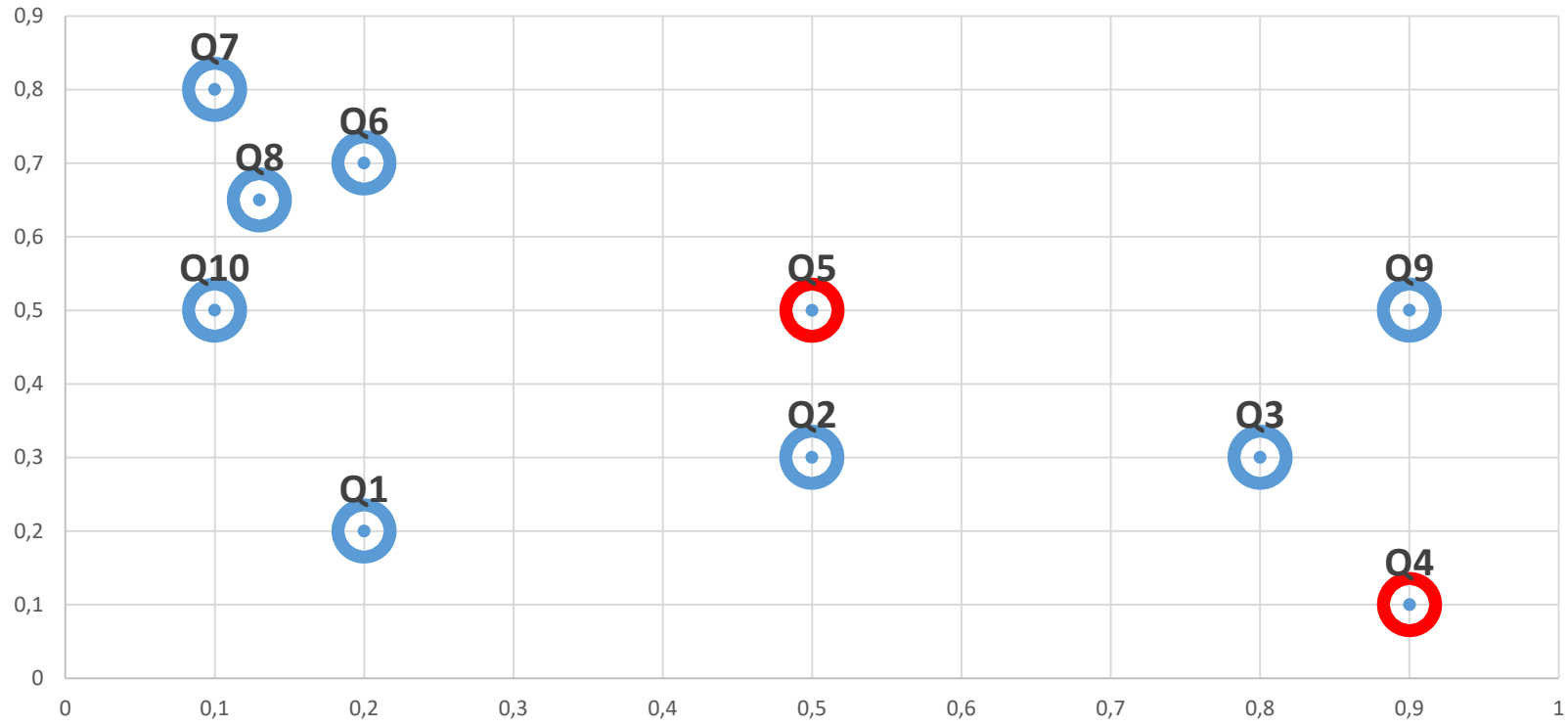



*Red is our first exemplar

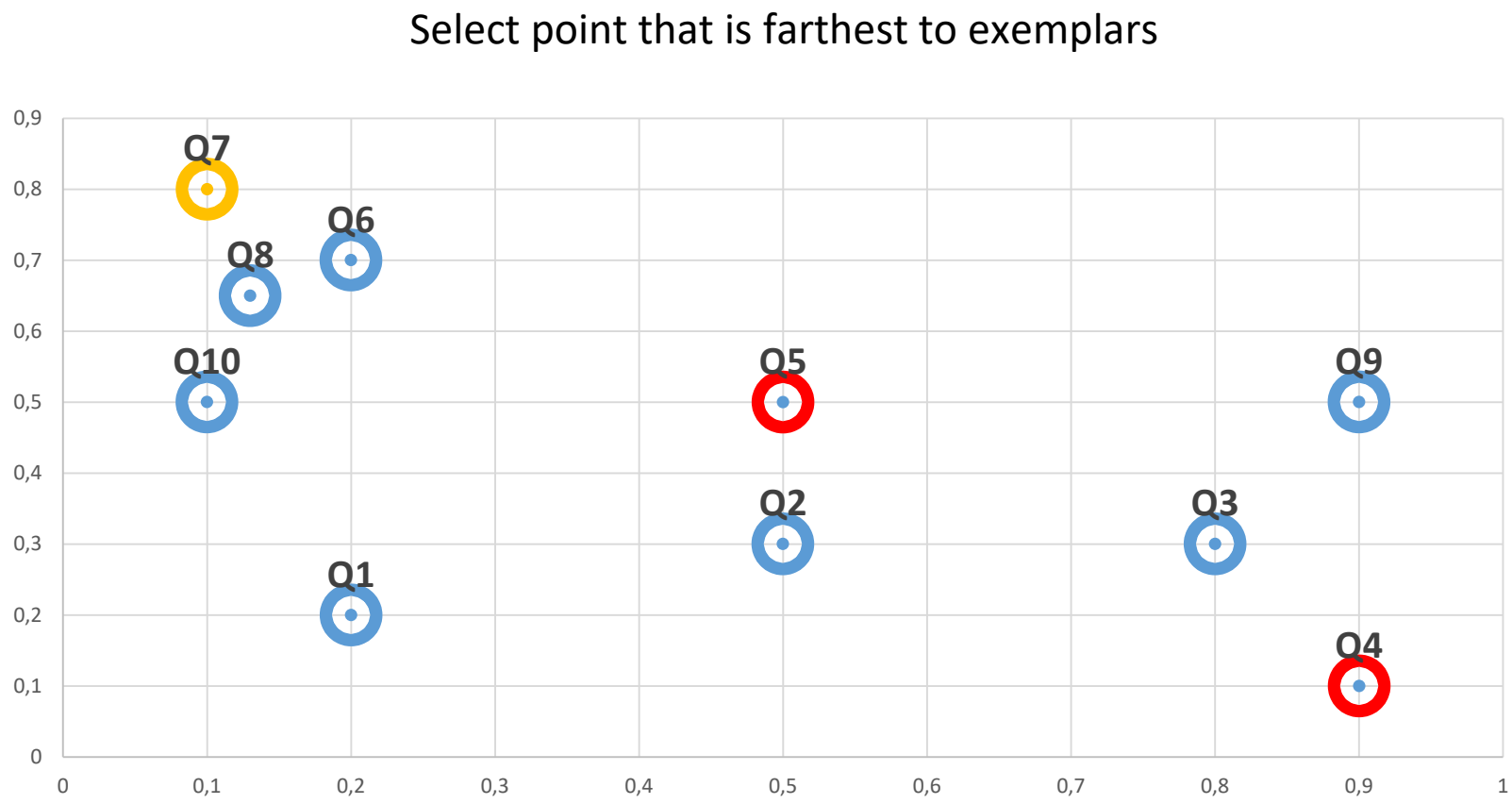
# Step3: Clusters Formation



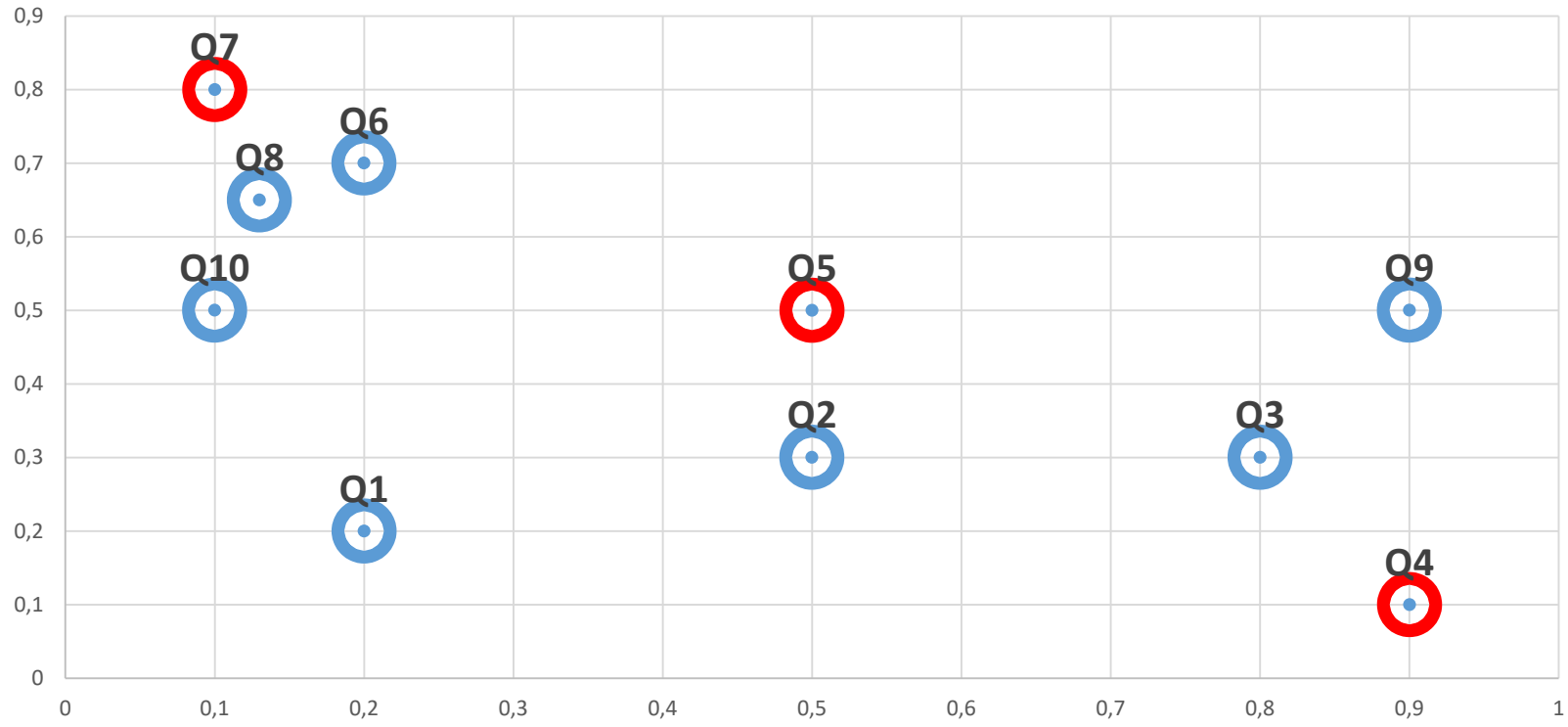
# Step3: Clusters Formation



# Step3: Clusters Formation

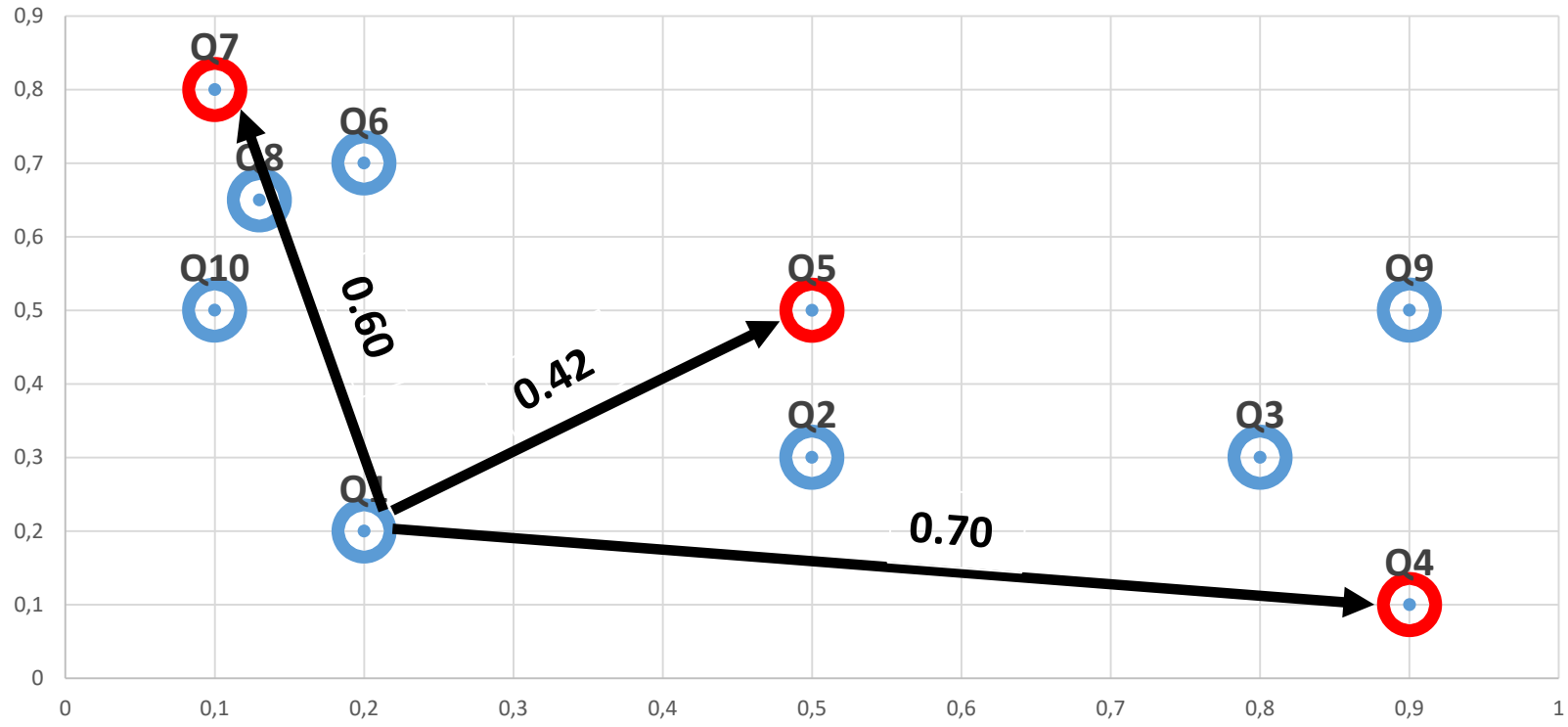


# Step3: Clusters Formation



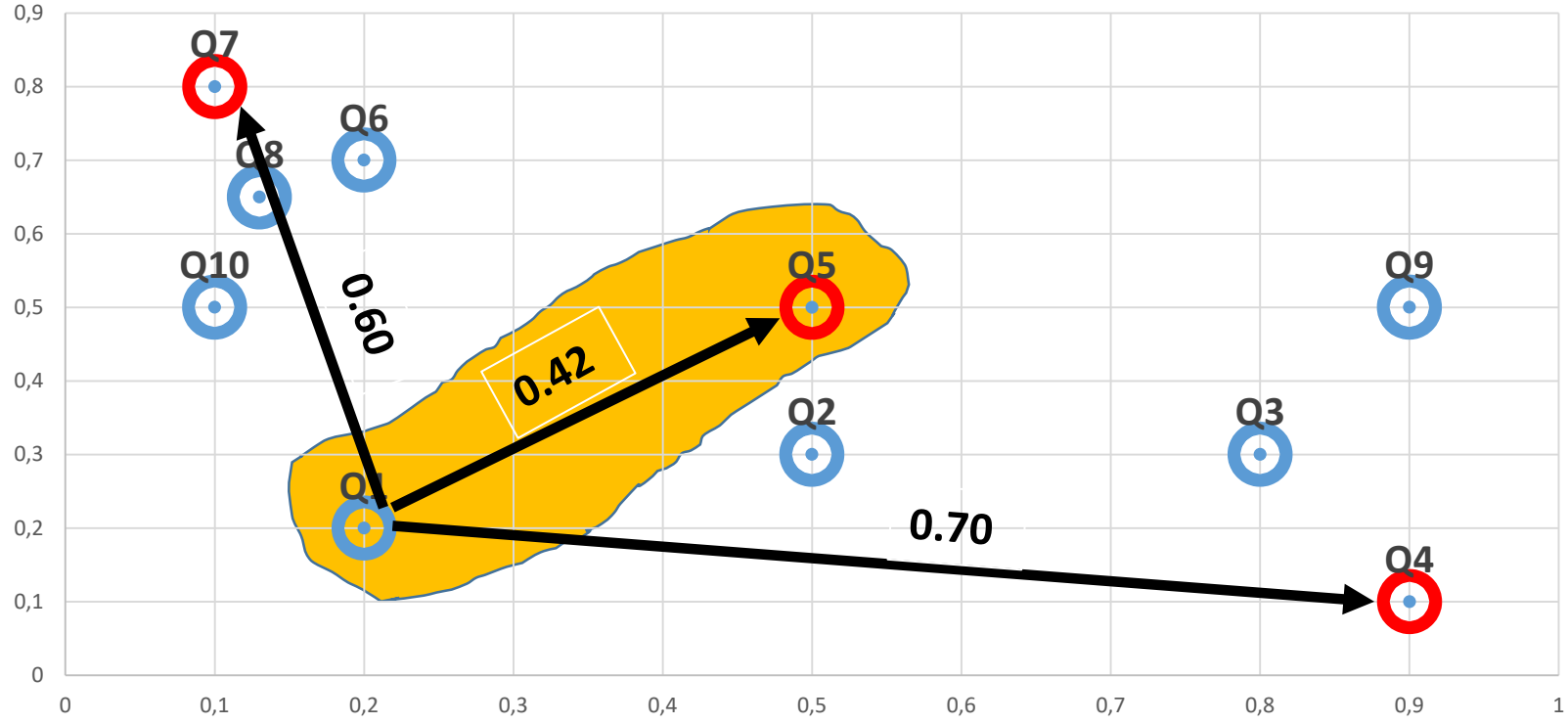
# Step3: Clusters Formation

Calculate distance from Q1 to each exemplars



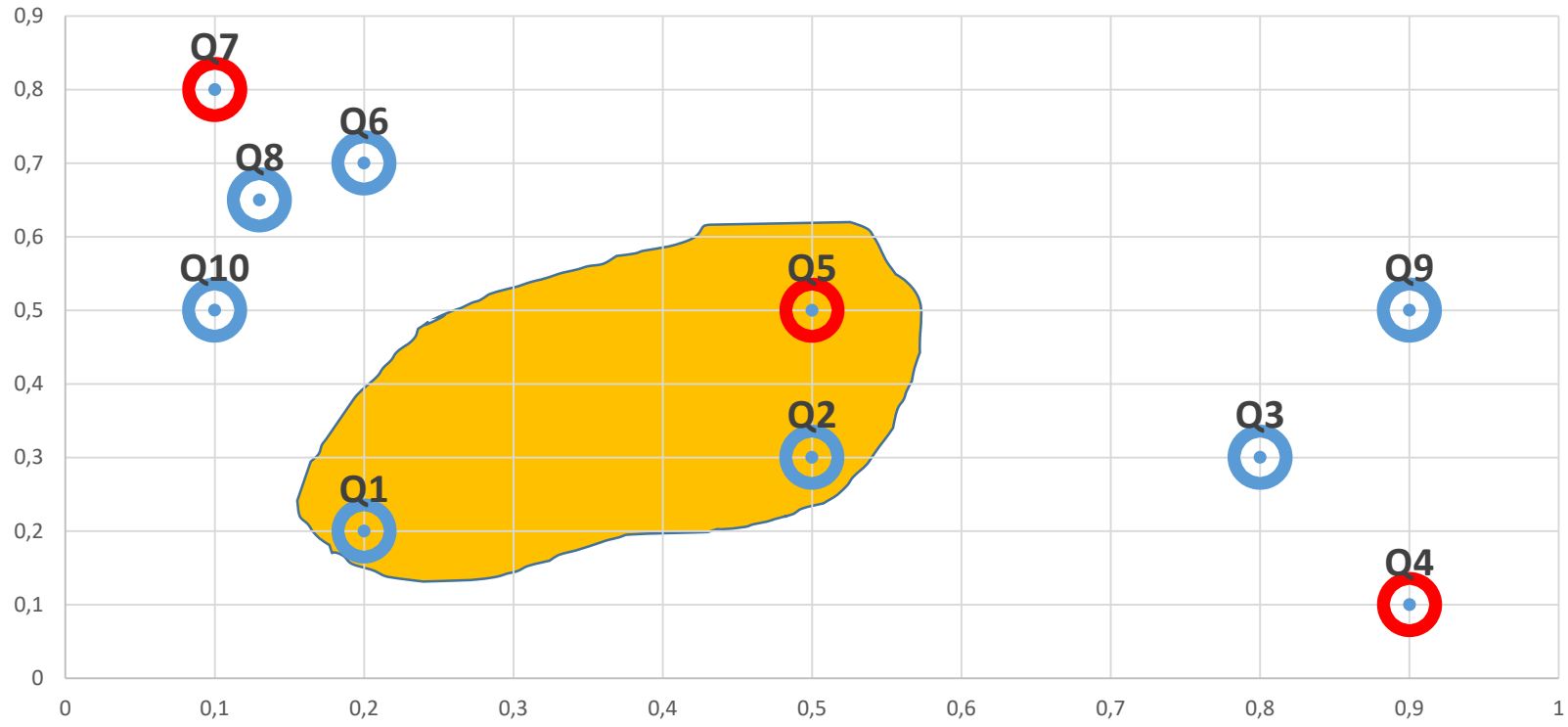
# Step3: Clusters Formation

Assign Q1 to the minimum distance exemplar



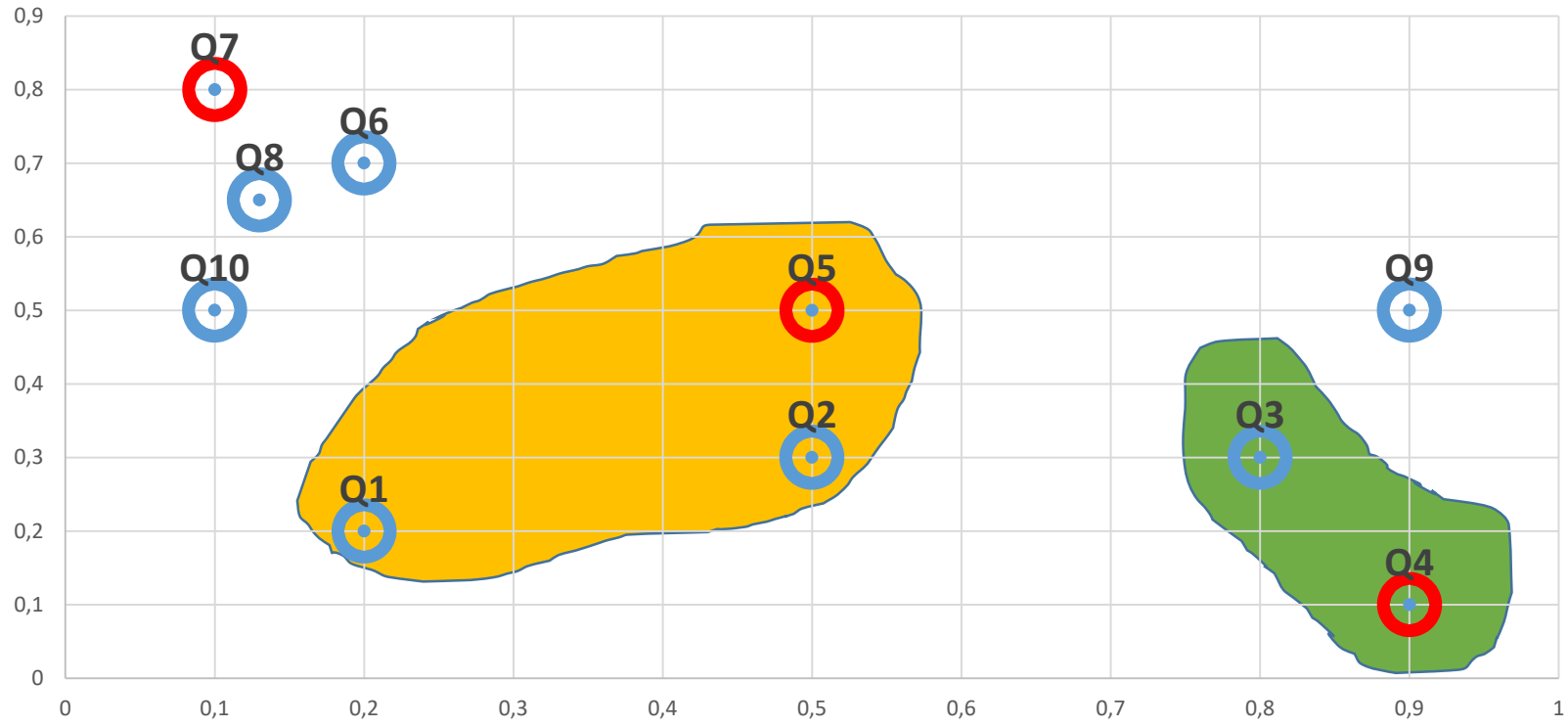
# Step3: Clusters Formation

Repeat the process for Q2



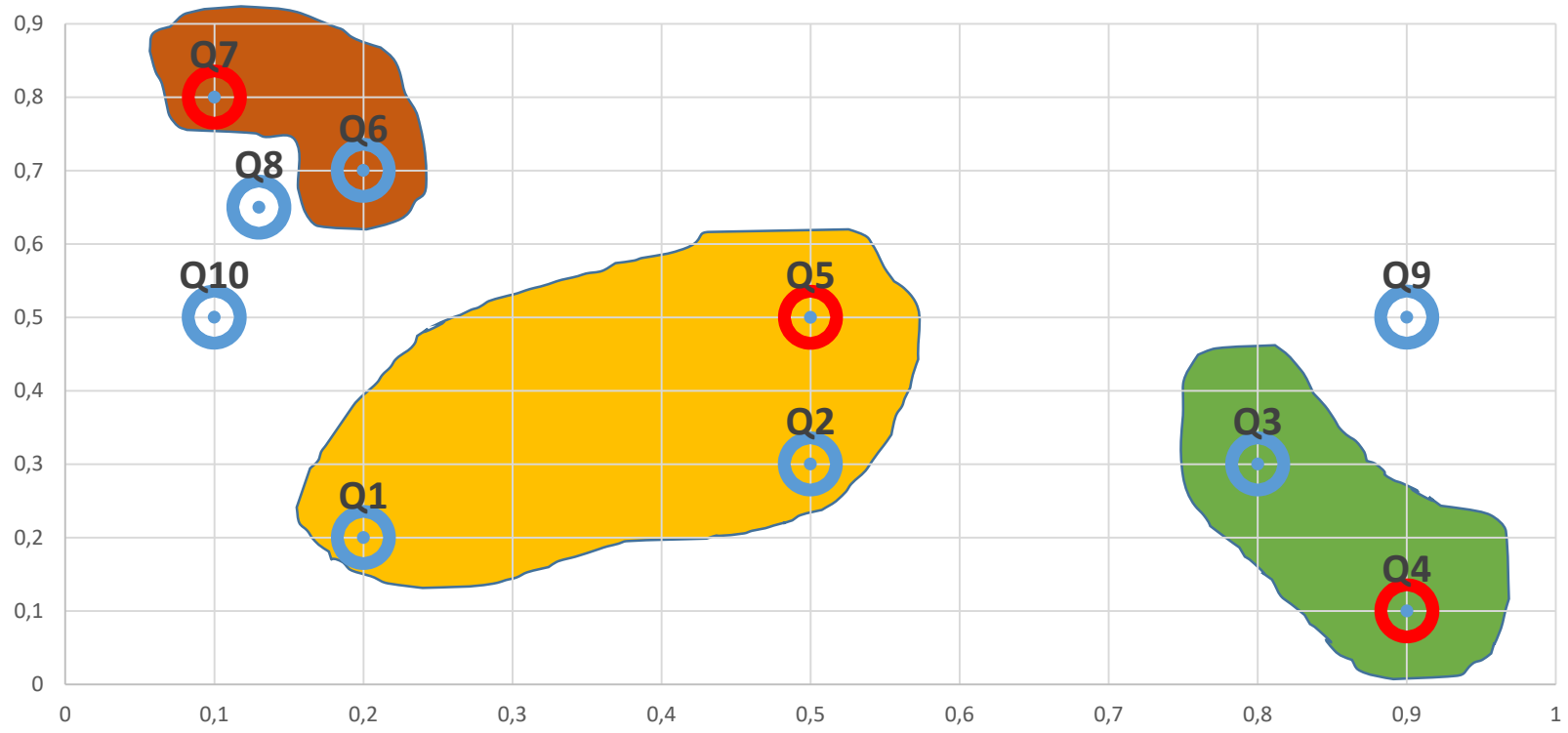
# Step3: Clusters Formation

Repeat the process for Q3



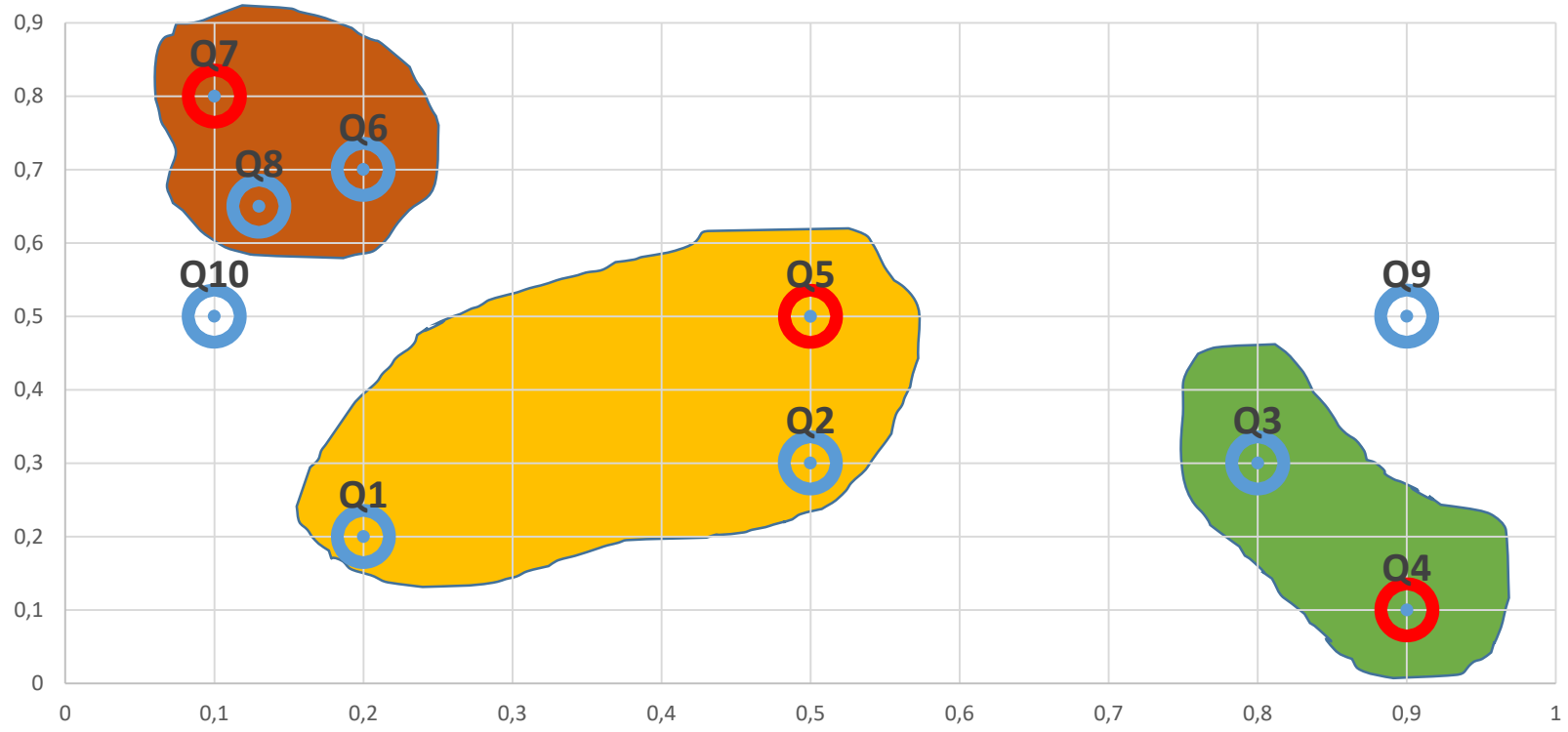
# Step3: Clusters Formation

Repeat the process for Q6



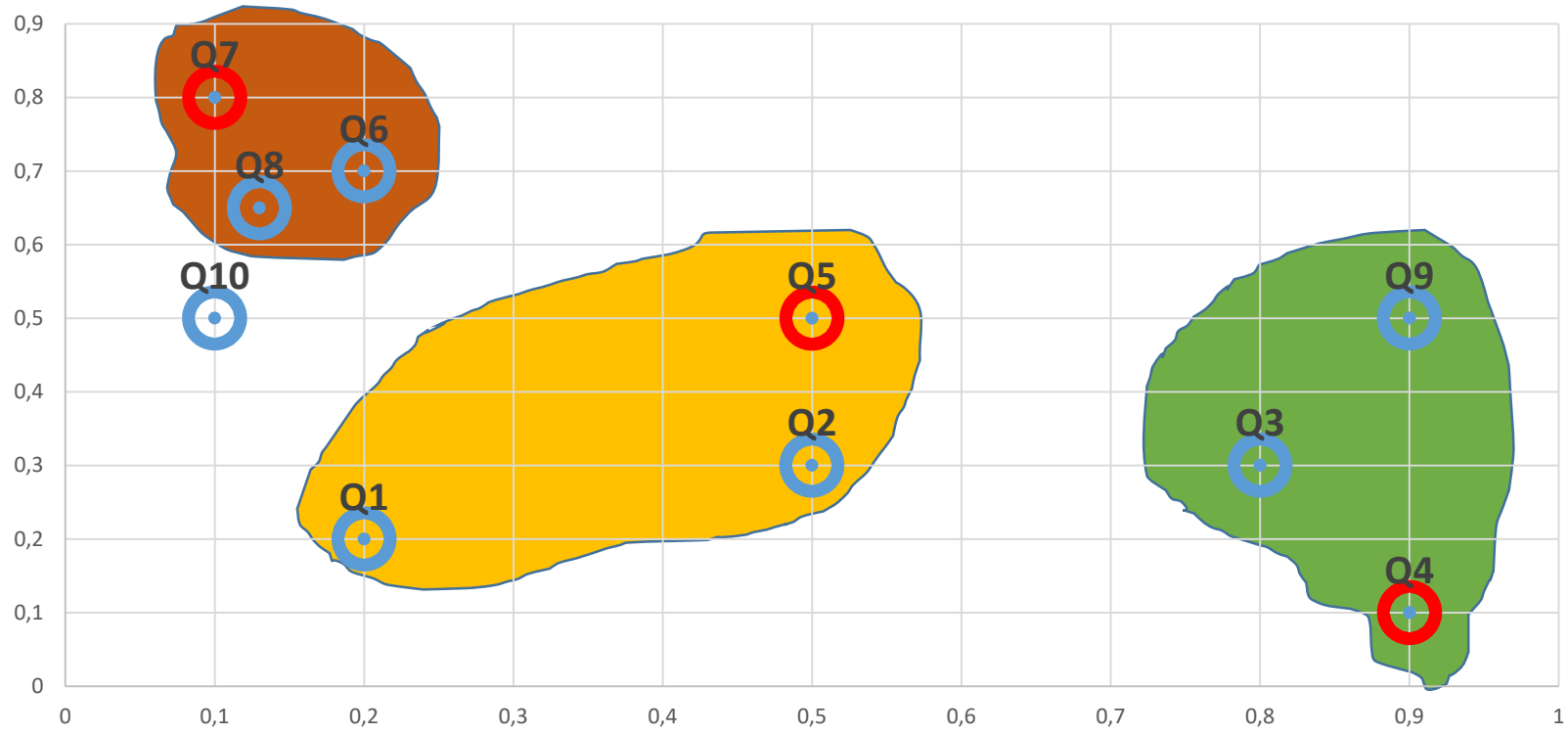
# Step3: Clusters Formation

Repeat the process for Q8



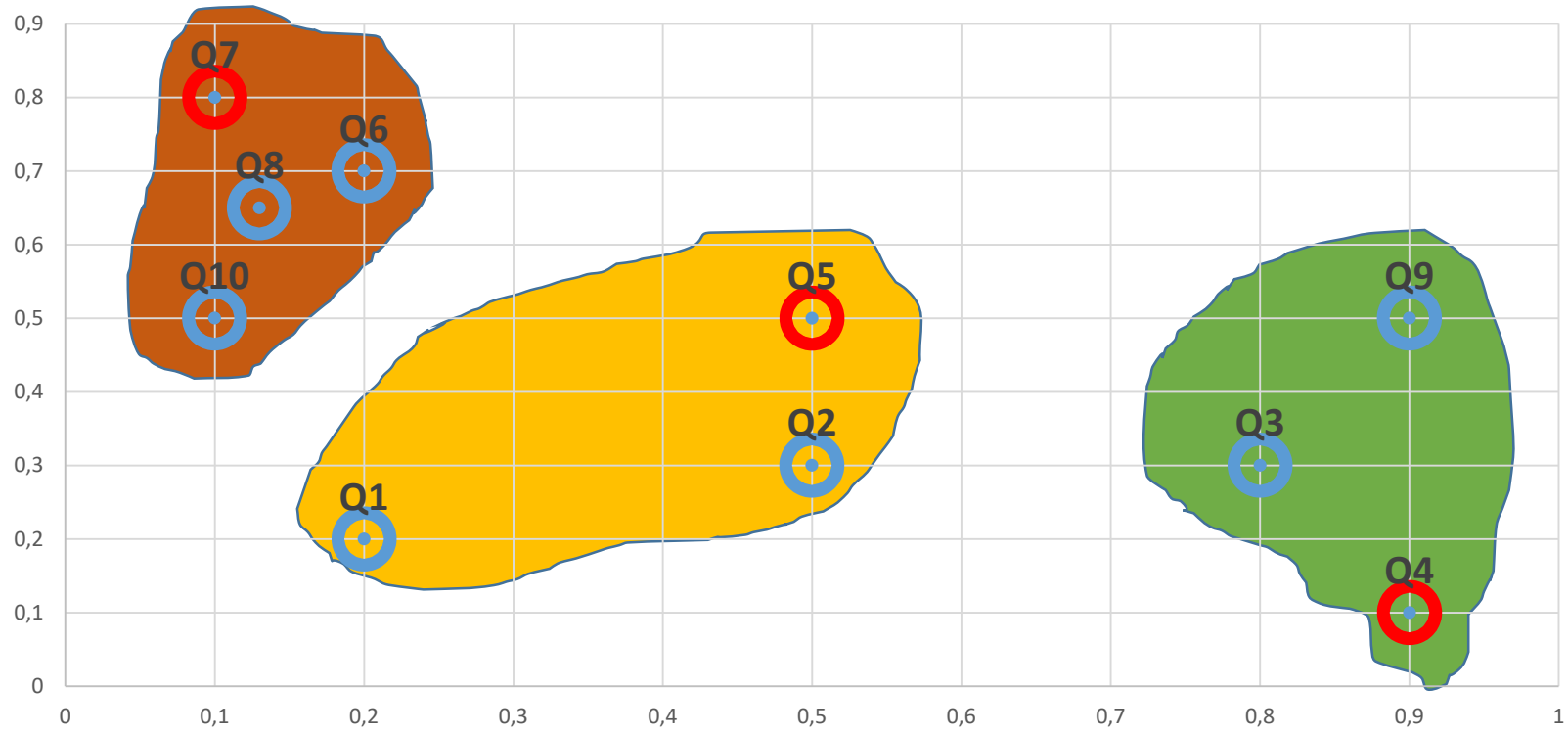
# Step3: Clusters Formation

Repeat the process for Q9

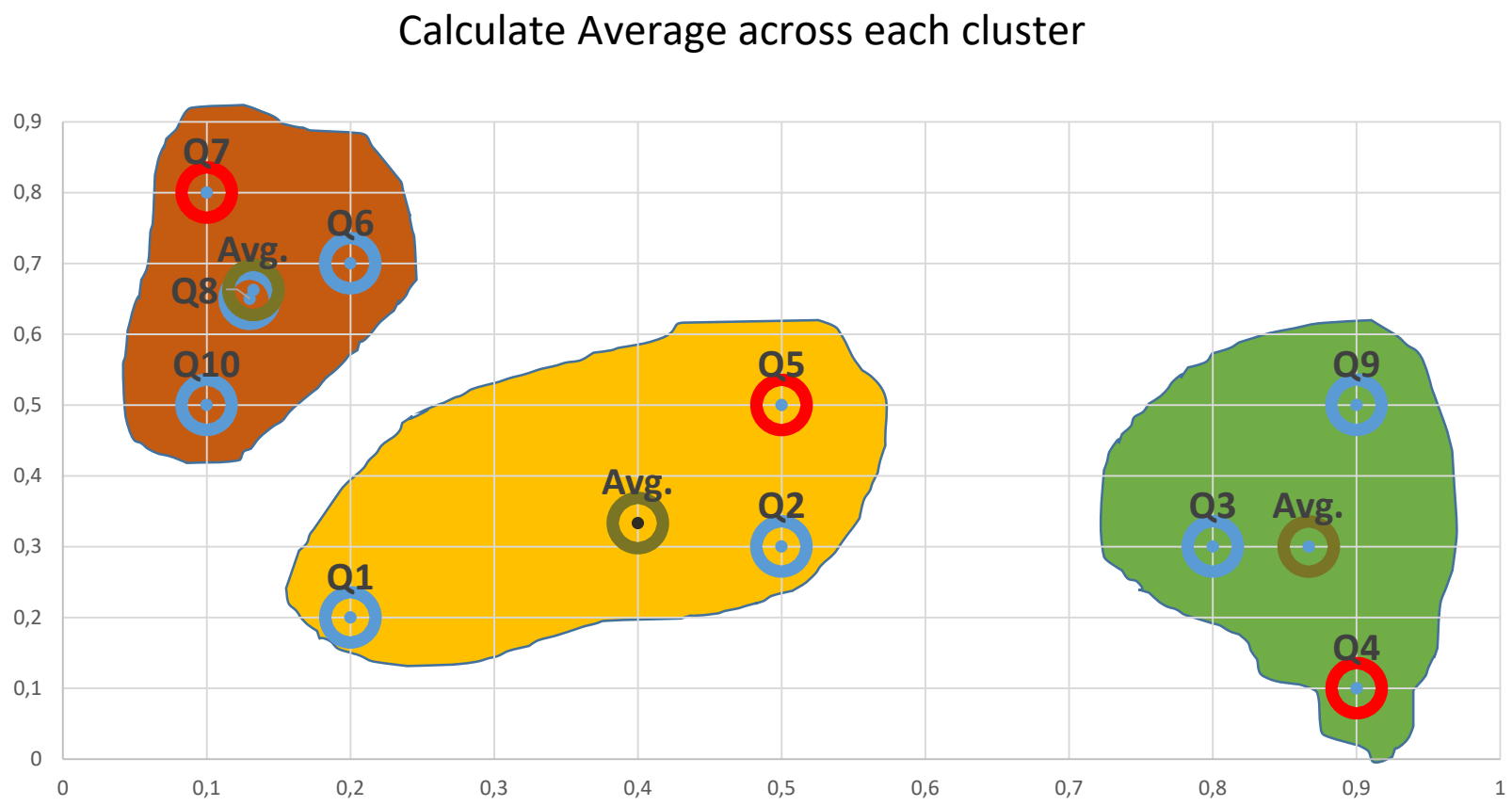


# Step3: Clusters Formation

Repeat the process for Q10

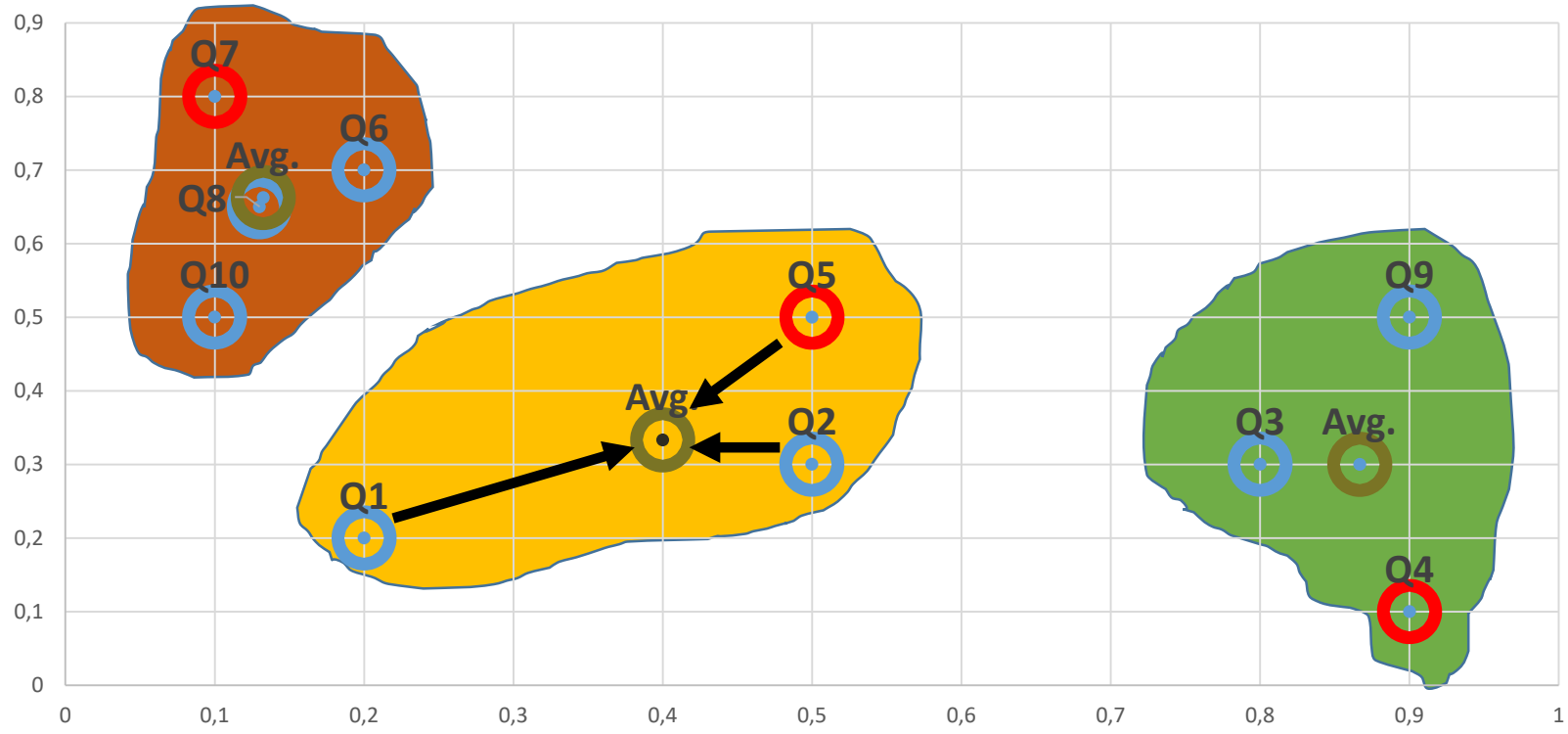


# Step4: Final Questions Selection



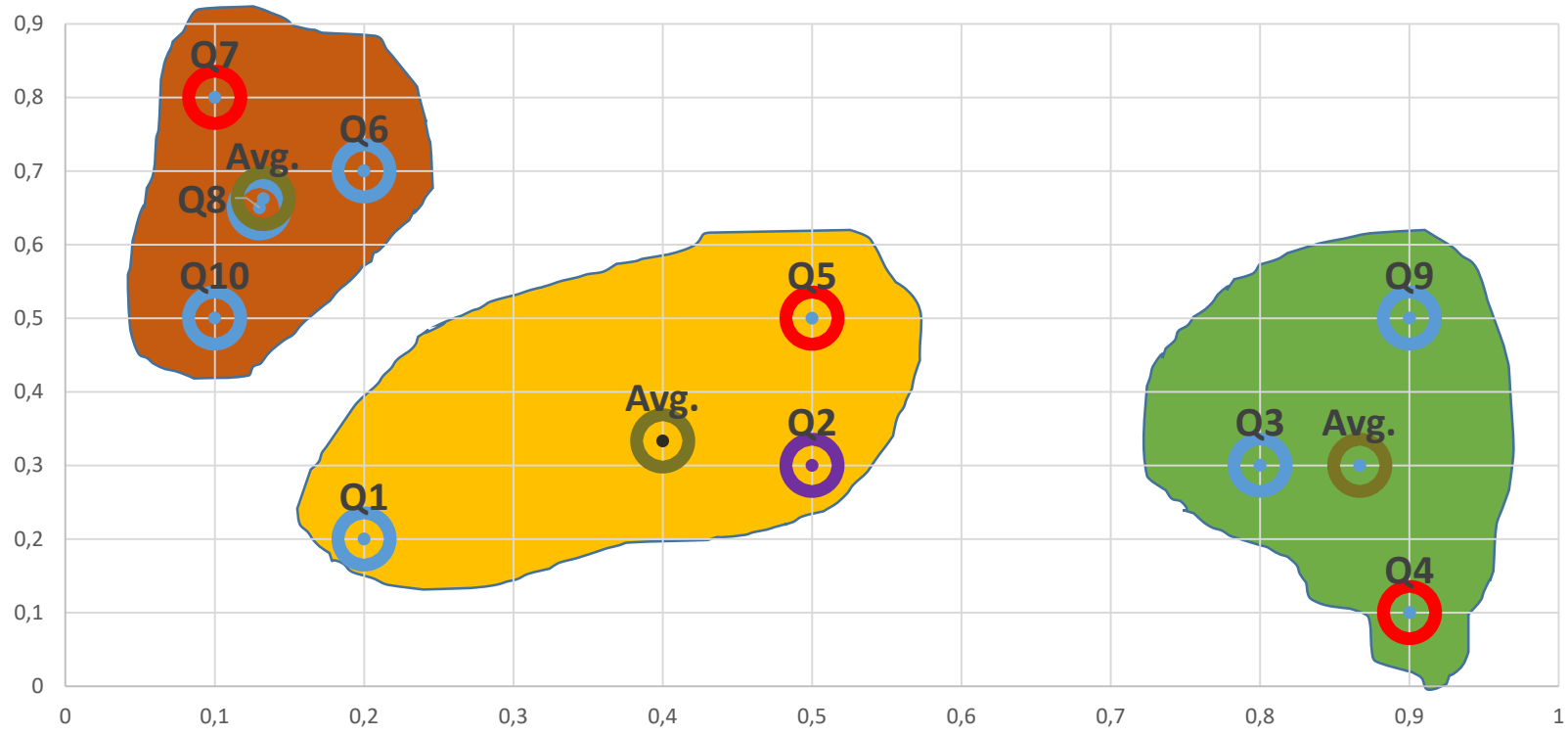
# Step4: Final Questions Selection

Calculate distance of each point in cluster to the average



# Step4: Final Questions Selection

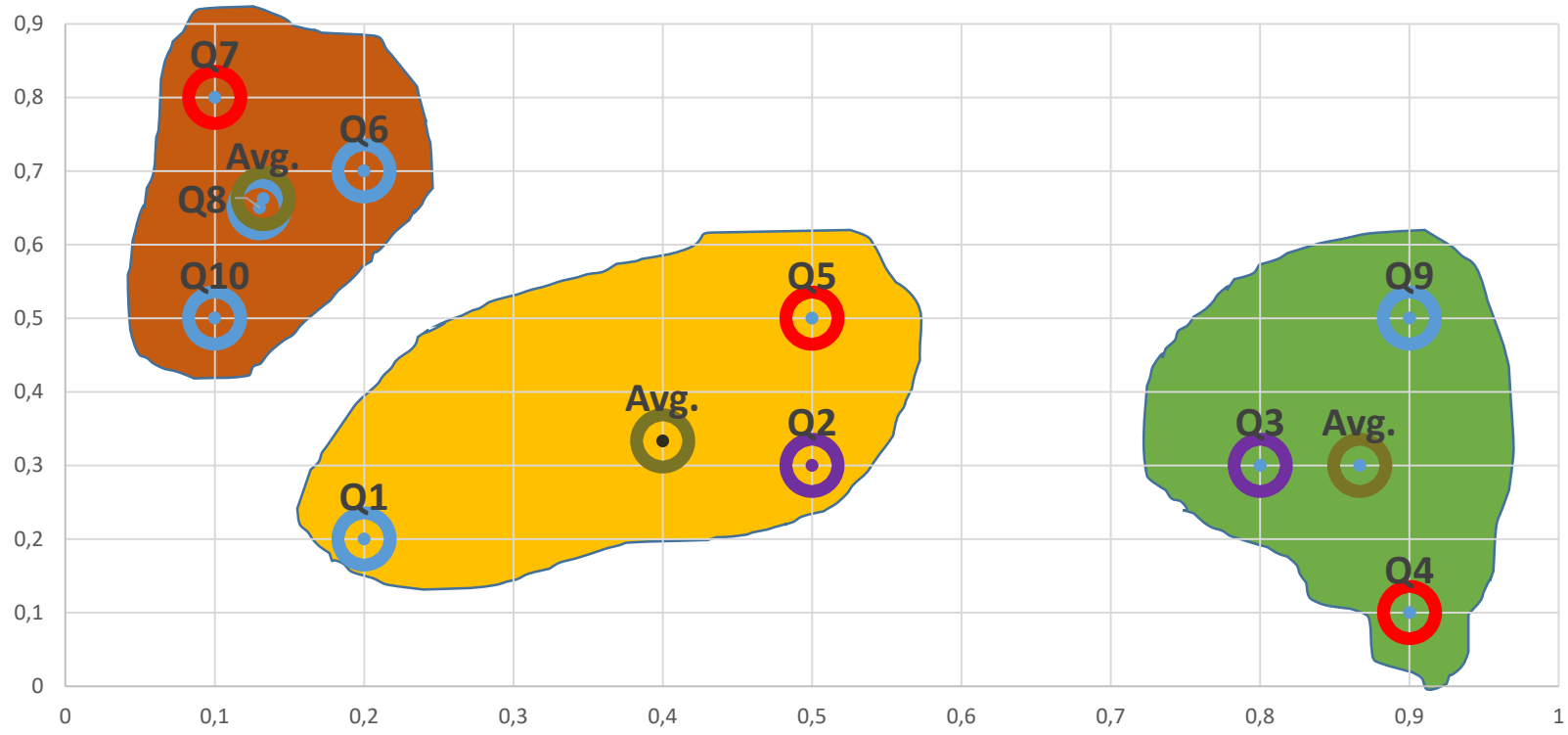
Select minimum distance question as the final benchmark query from that cluster




Purple, i.e., Q2 is the final selected question from yellow cluster

# Step4: Final Questions Selection

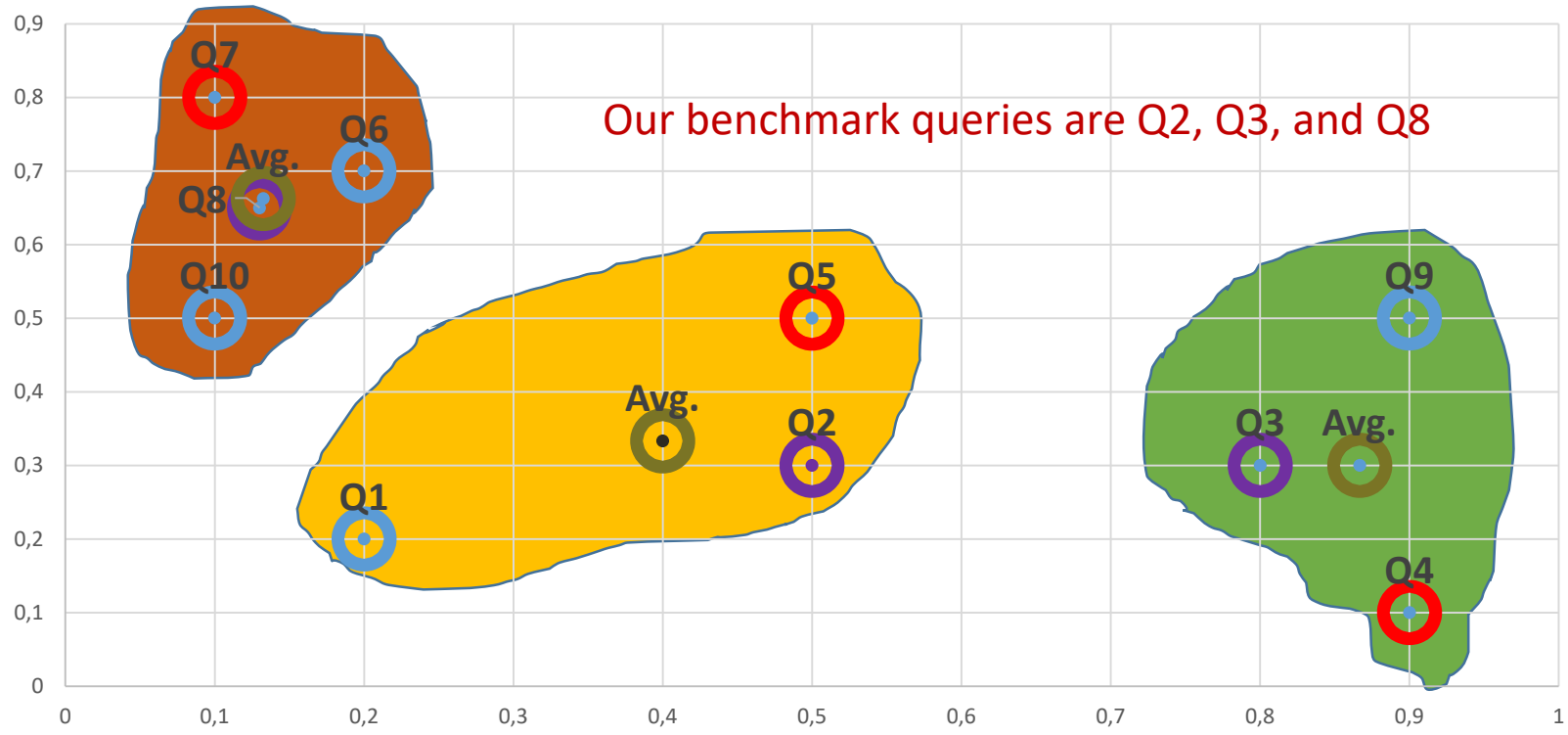
Select minimum distance query as the final benchmark query from that cluster




Purple, i.e., Q3 is the final selected question from green cluster

# Step4: Final Questions Selection

Select minimum distance query as the final benchmark query from that cluster




Purple, i.e., Q8 is the final selected question from brown cluster

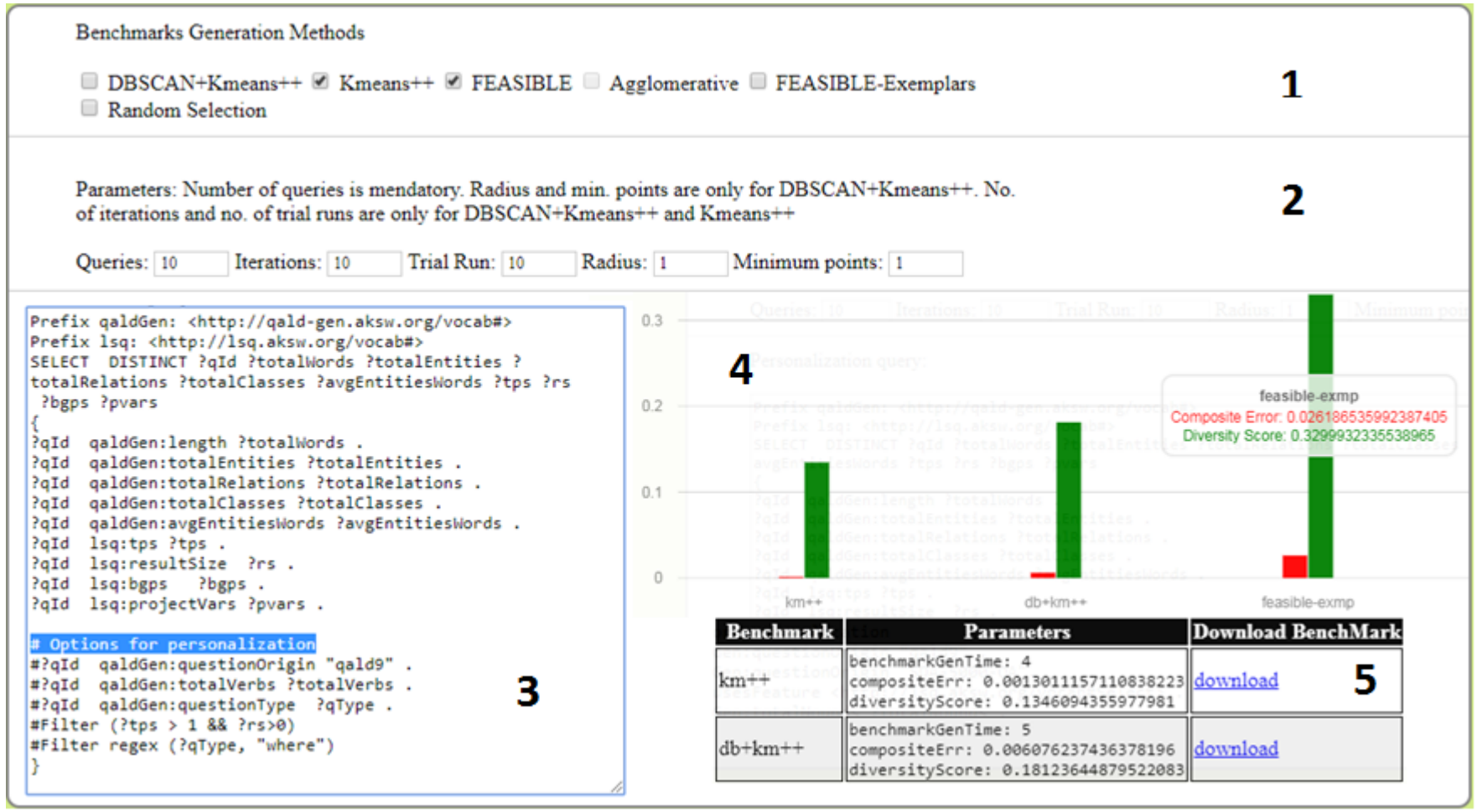
# Evaluation Setup

- GERBIL and Frankenstein frameworks are used to provide common experiment settings
- Personalised 200 question sample is generated using QaldGen for evaluating end to end QA system
- Personalised 100 question samples are generated using QaldGen for evaluating NED and Relation linking tools
- Precision, Recall, F-score are measured to report final performances.

# Results

| Systems | Dataset | Task | P | R | F |
|---|---|---|---|---|---|
| Baseline(gAnswer2) | QALD-9 | QA system | 0.29 | 0.33 | .030 |
| gAnswer2 | QaldGen | QA system | 0.24 | 0.24 | 0.24 |
| QUEPY | QaldGen | QA system | **0.27** | **0.27** | **0.27** |
| Baseline (TagMe) | LC-QuAD | NED | 0.69 | 0.67 | 0.68 |
| TagMe | QaldGen | NED | **0.44** | **0.40** | **0.41** |
| DBpedia Spotlight | QaldGen | NED | 0.37 | 0.38 | 0.36 |
| Baseline (RNLIWOD) | LC-QuAD | RL | 0.25 | 0.22 | 0.23 |
| RNLIWOD | QaldGen | RL | 0.23 | 0.19 | 0.20 |
| EARL | QaldGen | RL | **0.38** | **0.39** | **0.37** |

# QaldGen Demo (http://qaldgen.aksw.org/)



# Conclusions

- The definition of a “baseline” is subjected to the type of questions considered
  - Overall baseline on a dataset may not be a baseline for particular type of questions
  - Reason: A QA system or its component for modular framework may not be implemented to target some type of questions
- Microbenchmarking helps to understand exact pitfalls of a system
- For future
  - Integrate LCQuad 2.0
  - Google QA dataset

# Thanks

- QaldGen is open source https://github.com/dice-group/qald-generator
- ISWC 2019 Demo available from http://qaldgen.aksw.org/