



# Prompt Engineering – Final Presentation

Saqib Razzaq, Joey Hieronimy

# Outline

- Task description
- Dataset
- Prompting
- Comparisons
- Video Game Dataset
- Conclusion

# Task description

- Prompt Engineering
  - Prompt a LLM to get desired output
  - Refine prompt
  - Correct false outputs
  - Improve output format
- Different techniques
  - Zero-Shot
  - One-Shot
  - Few-Shot
  - Chain-Of-Thought
  - Self-Consistency
  - General-Knowledge

# Task description

## ■ Generating RML Mappings

```
{
  "characters": [
    {
      "id": "0",
      "firstname": "Ash",
      "lastname": "Ketchum",
      "hair": "black"
    },
    {
      "id": "1",
      "firstname": "Misty",
      "hair": "orange"
    }
  ]
}
```

```
@prefix rml: <http://semweb.mmlab.be/ns/rml#> .
@prefix rr: <http://www.w3.org/ns/r2rml#> .
@prefix ql: <http://semweb.mmlab.be/ns/ql#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix : <http://example.org/rules/> .
@prefix schema: <http://schema.org/> .
@prefix dbo: <http://dbpedia.org/ontology/> .

:TriplesMap a rr:TriplesMap;
  rml:logicalSource [
    rml:source "characters.json";
    rml:referenceFormulation ql:JSONPath;
    rml:iterator "$.characters[*]"
  ].

:TriplesMap rr:subjectMap [
  rr:template "http://example.org/character/{id}"
].

:TriplesMap rr:predicateObjectMap [
  rr:predicate rdf:type;
  rr:objectMap [
    rr:constant schema:Person
  ]
].

:TriplesMap rr:predicateObjectMap [
  rr:predicate schema:givenName;
  rr:objectMap [
    rml:reference "firstname"
  ]
].

:TriplesMap rr:predicateObjectMap [
  rr:predicate schema:lastName;
  rr:objectMap [
    rml:reference "lastname"
  ]
].

:TriplesMap rr:predicateObjectMap [
  rr:predicate dbo:hairColor;
  rr:objectMap [
    rml:reference "hair"
  ]
].
```

```
@prefix dbo: <http://dbpedia.org/ontology/> .
@prefix schema: <http://schema.org/> .

<http://example.org/character/0> a schema:Person;
  dbo:hairColor "black";
  schema:givenName "Ash";
  schema:lastName "Ketchum" .

<http://example.org/character/1> a schema:Person;
  dbo:hairColor "orange";
  schema:givenName "Misty" .
```

# Task description

## ■ Generating RML Mappings

```
{
  "characters": [
    {
      "id": "0",
      "firstname": "Ash",
      "lastname": "Ketchum",
      "hair": "black"
    },
    {
      "id": "1",
      "firstname": "Misty",
      "hair": "orange"
    }
  ]
}
```

```
@prefix rml: <http://semweb.mmlab.be/ns/rml#> .
@prefix rr: <http://www.w3.org/ns/r2rml#> .
@prefix ql: <http://semweb.mmlab.be/ns/ql#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix : <http://example.org/rules/> .
@prefix schema: <http://schema.org/> .
@prefix dbo: <http://dbpedia.org/ontology/> .

:TriplesMap a rr:TriplesMap;
  rml:logicalSource [
    rml:source "characters.json";
    rml:referenceFormulation ql:JSONPath;
    rml:iterator "$.characters[*]"
  ].

:TriplesMap rr:subjectMap [
  rr:template "http://example.org/character/{id}"
].

:TriplesMap rr:predicateObjectMap [
  rr:predicate rdf:type;
  rr:objectMap [
    rr:constant schema:Person
  ]
].

:TriplesMap rr:predicateObjectMap [
  rr:predicate schema:givenName;
  rr:objectMap [
    rml:reference "firstname"
  ]
].

:TriplesMap rr:predicateObjectMap [
  rr:predicate schema:lastName;
  rr:objectMap [
    rml:reference "lastname"
  ]
].

:TriplesMap rr:predicateObjectMap [
  rr:predicate dbo:hairColor;
  rr:objectMap [
    rml:reference "hair"
  ]
].
```

```
@prefix dbo: <http://dbpedia.org/ontology/> .
@prefix schema: <http://schema.org/> .

<http://example.org/character/0> a schema:Person;
  dbo:hairColor "black";
  schema:givenName "Ash";
  schema:lastName "Ketchum" .

<http://example.org/character/1> a schema:Person;
  dbo:hairColor "orange";
  schema:givenName "Misty" .
```

# Task description

- Prompt Engineering
  - Prompt a LLM to get desired output
  - Refine prompt
  - Correct false outputs
  - Improve output format
- Different techniques
  - Zero-Shot
  - One-Shot
  - Few-Shot
  - Chain-Of-Thought
  - Self-Consistency
  - General-Knowledge



# Dataset

- IMDb Top 1000 dataset
- Internet Movie Database
- Provide open dataset of Top 1000 rated films
- 16 data columns, 1000 instances

```
Poster_Link ,Series_Title ,Released_Year ,Certificate  
,Runtime ,Genre ,IMDB_Rating ,Overview ,Meta_score  
,Director ,Star1 ,Star2 ,Star3 ,Star4 ,No_of_Votes ,Gross
```



# Dataset

```
Poster_Link ,Series_Title ,Released_Year ,Certificate ,Runtime ,Genre  
,IMDB_Rating ,Overview ,Meta_score ,Director ,Star1 ,Star2 ,Star3 ,Star4  
,No_of_Votes ,Gross
```

```
"https://m.media-  
amazon.com/images/M/MV5BMDFkYTc0MGEtZmNhMC00ZDIzLWFmNTEtODM1ZmRlYWMwMWFmXkEyXkFqc  
GdeQXVyMTMxODk2OTU@._V1_UX67_CR0,0,67,98_AL_.jpg" ,The Shawshank Redemption ,1994  
,A ,142 min ,Drama , 9.3 , "Two imprisoned men bond over a number of years,  
finding solace and eventual redemption through acts of common decency." , 80  
,Frank Darabont ,Tim Robbins ,Morgan Freeman ,Bob Gunton ,William Sadler ,  
2343110 , "28,341,469"
```



# Prompting

- ChatGPT 3.5
- Chat Generative Pre-trained Transformer based on a LLM
- Developed by OpenAI
- Free and widely available to everyone
- Prompting techniques from the Prompt Engineering Guide<sup>1</sup>
- RDF Generation
  - Zero-Shot
  - One-Shot
- RML Generation
  - Zero-Shot
  - One-Shot
  - Few-Shot



# RDF Generation – One-Shot

- Provided an example output additionally to instruction
- Clearly separated instruction from Task

Convert the following CSV into RDF:

Poster_Link					,Series_Title
,IMDB_Rating					,Overview
,Meta_score	,Director	,Star1	,Star2	,Star3	,Star4
"https://m.media-					
amazon.com/images/M/MV5BMDFkYkYtOMGEtZmNhMC00ZDZlZWJmNTFmODM1ZmRlYWMy					
,The Shawshank Redemption		,1994	,A	,142 min	,Drama
and eventual redemption through acts of common decency."					
Darabont	,Tim Robbins	,Morgan Freeman	,Bob Gunton		
					,William Sa

•

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

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

@prefix ex: <http://example.org/film#>.

```
ex:shawshankRedemption
  a ex:Movie ;
  ex:posterLink "https://m.media-
amazon.com/images/M/MV5BMDFkYTc0MGEtZmNhMC00ZDIzLWFmNTEtODM1ZmRIYWVv
;
  ex:seriesTitle "The Shawshank Redemption";
```

```
ex:releasedYear 1994 ;
ex:certificate "A" ;
ex:runtime "142 min" ;
ex:genre "Drama" ;
ex:imdbRating 9.3 ;
ex:overview "Two imprisoned men bond over a number of years, finding solace and eventual i
ex:metaScore 80 ;
ex:director "Frank Darabont" ;
ex:star1 "Tim Robbins" ;
ex:star2 "Morgan Freeman" ;
ex:star3 "Bob Gunton" ;
ex:star4 "William Sadler" ;
ex:noOfVotes 2343110 ;
ex:gross "28,341,469" .
```

###

"https://m.media-  
amazon.com/images/M/MV5BM2MyNjYxNmUtYTawNi00MTYxLWJmNWYtYzZIODY3ZTk3OT  
The Godfather ,1972 ,A ,175 min ,"Crime, Drama" ,  
clandestine empire to his reluctant son.  
,Marlon Brando ,Al Pacino ,James Caan ,Diane Keaton , 1620367 ,  
"https://m.media-amazon.com/images/M/MV5BMTMxNTMwODM0NF5BMl5BanBnXkFtZW  
Dark Knight ,2008 ,UA ,152 min ,"Action, Crime, Drama" ,  
people of Gotham, Batman must accept one of the greatest psychological and physical tests of  
, 84 ,Christopher Nolan ,Christian Bale ,Heath Ledger ,Aaron Eckhart  
:

# RDF Generation – One-Shot

```
ex:godfather
  a ex:Movie ;
  ex:posterLink "https://m.media-amazon.com/images/M/MV5BM2MyNjYxN
  ex:seriesTitle "The Godfather" ;
  ex:releasedYear 1972 ;
  ex:certificate "A" ;
  ex:runtime "175 min" ;
  ex:genre "Crime, Drama" ;
  ex:imdbRating 9.2 ;
  ex:overview "An organized crime dynasty's aging patriarch transf
  ex:metaScore 100 ;
  ex:director "Francis Ford Coppola" ;
  ex:star1 "Marlon Brando" ;
  ex:star2 "Al Pacino" ;
  ex:star3 "James Caan" ;
  ex:star4 "Diane Keaton" ;
  ex:noOfVotes 1620367 ;
  ex:gross "134,966,411" .
```

```
ex:darkKnight
  a ex:Movie ;
  ex:posterLink "https://m.media-amazon.com/images/M/MV5BMTMxNTMwO
  ex:seriesTitle "The Dark Knight" ;
  ex:releasedYear 2008 ;
  ex:certificate "UA" ;
  ex:runtime "152 min" ;
  ex:genre "Action, Crime, Drama" ;
  ex:imdbRating 9.0 ;
  ex:overview "When the menace known as the Joker wreaks havoc and
  ex:metaScore 84 ;
  ex:director "Christopher Nolan" ;
  ex:star1 "Christian Bale" ;
  ex:star2 "Heath Ledger" ;
  ex:star3 "Aaron Eckhart" ;
  ex:star4 "Michael Caine" ;
  ex:noOfVotes 2303232 ;
  ex:gross "534,858,444" .
```

# RML Generation

- Now generating RML
- RDF Mapping Language
- Used to generate RDF out of structured data (JSON, CSV, XML,...)
- Portability - one RML file can be used on a lot of data
- Define set of mapping rules
- Consists of Logical Source, Subject Map and zero or more Predicate-Object Maps

```
<#PersonMapping>
  rml:logicalSource [
    rml:source "People.json";
    rml:referenceFormulation ql:JSONPath;
    rml:iterator "$.[*].People" ].
```

```
<#PersonMapping>
  rr:subjectMap [
    rr:template "http://ex.com/Person/{name}_{surname}";
    rr:class ex:Person ].
```

```
<#PersonMapping>
  rr:predicateObjectMap [
    rr:predicate ex:name;
    rr:objectMap [ rml:reference "name" ] ].
```

# RML – Zero-Shot

- First attempt: generate RML with no info



## Anonymous

I have a dataset containing Information about films. Generate a RML file to convert them into RDF.

rml

Copy code

```
@prefix rr: <http://www.w3.org/ns/r2rml#> .
@prefix rml: <http://semweb.mmlab.be/ns/rml#> .
@prefix ql: <http://semweb.mmlab.be/ns/ql#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<#FilmMapping>
rr:logicalTable [
  rml:source "your-film-dataset.csv";
  rml:referenceFormulation ql:CSV
];

rr:subjectMap [
  rr:template "http://example.org/films/{Title}";
  rr:class <http://example.org/ontology/Film>
];

rr:predicateObjectMap [
  rr:predicate <http://example.org/ontology/title>;
  rr:objectMap [ rr:column "Title" ]
];

rr:predicateObjectMap [
  rr:predicate <http://example.org/ontology/director>;
  rr:objectMap [ rr:column "Director" ]
];

rr:predicateObjectMap [
  rr:predicate <http://example.org/ontology/releaseYear>;
  rr:objectMap [ rr:column "ReleaseYear"; rr:datatype xsd:integer ]
].
```

# RML – Zero-Shot

- Result improved after providing more details
- No schema prefix!



**Anonymous**

These are the columns of my data:

Poster_Link					
,Series_Title		,Released_Year	,Certificate	,Runtime	,Genre
,IMDB_Rating	,Overview				
,Meta_score	,Director	,Star1	,Star2	,Star3	,Star4
,No_of_Votes	,Gross				



**Anonymous**

Make sure to map them to the corresponding schema.org ontologies

rml

Copy code

```
@prefix rr: <http://www.w3.org/ns/r2rml#> .
@prefix rml: <http://semweb.mmlab.be/ns/rml#> .
@prefix ql: <http://semweb.mmlab.be/ns/ql#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<#FilmMapping>
  rr:logicalTable [
    rml:source "your-film-dataset.csv";
    rml:referenceFormulation ql:CSV
  ];

  rr:subjectMap [
    rr:template "http://example.org/films/{Series_Title}";
    rr:class <http://schema.org/Movie>
  ];

  rr:predicateObjectMap [
    rr:predicate <http://schema.org/name>;
    rr:objectMap [ rr:column "Series_Title" ]
  ];

  rr:predicateObjectMap [
    rr:predicate <http://schema.org/reviewCount>;
    rr:objectMap [ rr:column "No_of_Votes"; rr:datatype xsd:integer ]
  ];

  rr:predicateObjectMap [
    rr:predicate <http://schema.org/boxOffice>;
    rr:objectMap [ rr:column "Gross" ]
  ].
```



# RML – Zero-Shot

- Next only example data given
- Didn't generate the entire file
- No schema



Anonymous

Generate a RML mapping for this data:

"https://m.media-  
amazon.com/images/M/MV5BMDFkYTc0MGEtZmNhMC00ZDIzLWFmNTetODM1ZmRIYWw  
,The Shawshank Redemption,1994,A,142 min,Drama  
and eventual redemption through acts of common decency."  
Darabont,Tim Robbins,Morgan Freeman,Bob Gunton,William S

turtle

Copy code

```
@prefix rr: <http://www.w3.org/ns/r2rml#> .  
@prefix rml: <http://semweb.mmlab.be/ns/rml#> .  
@prefix ql: <http://semweb.mmlab.be/ns/ql#> .  
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
```

```
# Logical Source
```

```
<#LogicalSource>
```

```
  rml:source "your_data_source.csv";  
  rml:referenceFormulation ql:CSV.
```

```
# Subject Map
```

```
<#SubjectMap>
```

```
  rr:subjectMap [  
    rr:template "http://example.org/movies/{Title}";  
    rr:class ex:Movie;  
  ].
```

```
# PredicateObject Maps
```

```
<#Title>
```

```
  rr:predicate ex:title;  
  rr:objectMap [  
    rr:reference "The Shawshank Redemption";  
  ].
```

```
<#Year>
```

```
  rr:predicate ex:year;  
  rr:objectMap [  
    rr:reference "1994";  
    rr:datatype xsd:integer;  
  ].
```

```
<#Rating>
```

```
  rr:predicate ex:rating;  
  rr:objectMap [  
    rr:reference "9.3";  
    rr:datatype xsd:decimal;  
  ].
```

```
# ... (similar mappings for other properties)
```

```
# Triples Maps
```

```
<#TriplesMap>
```

```
  rr:logicalSource <#LogicalSource>;  
  rr:subjectMap <#SubjectMap>;  
  rr:predicateObjectMap <#Title>, <#Year>, <#Rating>, ... .
```

```
# Vocabulary Definitions (replace with your actual vocabulary)  
ex: Movie, title, year, rating, ... .
```

# RML – Zero-Shot

- Finally, all information provided:



**Anonymous**

Create an RML mapping that is able to convert CSV data with the following structure into a cc fitting schema.org ontology.

Poster_Link					,Series_
,IMDB_Rating	,Overview				
,Meta_score	,Director	,Star1	,Star2	,Star3	,Star4
"https://m.media-					
amazon.com/images/M/MV5BMDFkYTc0MGEtZmNhMC00ZDIzLWFmNTEtODM1ZmRIYW					
,The Shawshank Redemption		,1994	,A	,142 min	,Drama
and eventual redemption through acts of common decency."					
Darabont	,Tim Robbins	,Morgan Freeman	,Bob Gunton	,William S	

Copy code

```
@prefix rr: <http://www.w3.org/ns/r2rml#> .
@prefix rml: <http://semweb.mmlab.be/ns/rml#> .
@prefix ql: <http://semweb.mmlab.be/ns/ql#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix schema: <http://schema.org/> .
```

```
<#CSVMapping>
  a rr:TriplesMap;
  rr:logicalTable [
    rr:tableName "your_csv_file.csv";
  ];

  rr:subjectMap [
    rr:template "http://example.org/films/{Series_Title}";
    rr:class schema:Movie;
  ];

  rr:predicateObjectMap [
    rr:predicate schema:name;
    rr:objectMap [
      rr:column "Series_Title";
    ];
  ];

  rr:predicateObjectMap [
    rr:predicate schema:datePublished;
    rr:objectMap [
      rr:column "Released_Year";
      rr:datatype xsd:integer;
    ];
  ];
];
```

# RML – One-Shot

- Provide an example desired output RDF
- Generate RML based on that



## Anonymous

Create a RML file to map my CSV film dataset onto RDF. The RDF should later look like this:

```
@prefix schema: <http://schema.org/> .
```

```
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
```

```
<https://m.media-
```

```
amazon.com/images/M/MV5BMDFKYTc0MGEtZmNhMC00ZDIzLWFmNTEtODM1ZmRIYWw
```

```
a schema:Movie ;
```

```
schema:image "https://m.media-
```

```
amazon.com/images/M/MV5BMDFKYTc0MGEtZmNhMC00ZDIzLWFmNTEtODM1ZmRIYWw
```

```
;
```

```
schema:name "The Shawshank Redemption" ;
```

```
schema:datePublished "1994"^^xsd:gYear ;
```

```
schema:contentRating "A" ;
```

```
schema:duration "PT142M" ;
```

```
schema:genre "Drama" ;
```

rml

Copy code

```
@prefix rr: <http://www.w3.org/ns/r2rml#> .  
@prefix rml: <http://semweb.mmlab.be/ns/rml#> .  
@prefix ql: <http://semweb.mmlab.be/ns/ql#> .  
@prefix schema: <http://schema.org/> .  
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
```

```
<#TriplesMapFilm>
```

```
a rr:TriplesMap ;
```

```
rml:logicalSource [
```

```
    rml:source "your_film_dataset.csv";
```

```
    rml:referenceFormulation ql:CSV
```

```
] ;
```

```
rr:subjectMap [
```

```
    rr:template "https://m.media-amazon.com/images/{Image}" ;
```

```
    rr:class schema:Movie
```

```
] ;
```

```
rr:predicateObjectMap [
```

```
    rr:predicate schema:image ;
```

```
    rr:objectMap [
```

```
        rml:reference "Image"
```

```
    ]
```

```
] ;
```

# RML – One-Shot

```
<#TriplesMapReview>
a rr:TriplesMap ;
rr:subjectMap [
  rr:template "https://m.media-amazon.com/images/{Image}/review" ;
  rr:class schema:Review
] ;
rr:predicateObjectMap [
  rr:predicate schema:reviewRating ;
  rr:objectMap [
    rr:parentTriplesMap <#TriplesMapRating>
  ]
] .

<#TriplesMapPersonDirector>
a rr:TriplesMap ;
rr:subjectMap [
  rr:template "https://m.media-amazon.com/images/{Image}/director" ;
  rr:class schema:Person
] ;
rr:predicateObjectMap [
  rr:predicate schema:name ;
  rr:objectMap [
    rml:reference "Director"
  ]
] .

<#TriplesMapPersonActor>
a rr:TriplesMap ;
rr:subjectMap [
  rr:template "https://m.media-amazon.com/images/{Image}/actor" ;
  rr:class schema:Person
] ;
rr:predicateObjectMap [
  rr:predicate schema:name ;
  rr:objectMap [
    rml:reference "Actor"
  ]
] .
```

# RML – Few-Shot

- Provided two example RDFs



## Anonymous

Create a RML file to map data from a film dataset onto RDF. The output sho

```
@prefix schema: <http://schema.org/> .
```

```
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
```

```
<https://m.media-  
amazon.com/images/M/MV5BMDFkYTc0MGEtZmNhMC00ZDZlZWFnNT  
a schema:Movie ;  
schema:image "https://m.media-  
amazon.com/images/M/MV5BMDFkYTc0MGEtZmNhMC00ZDZlZWFnNT  
;  
schema:name "The Shawshank Redemption" ;
```

```
<https://m.media-  
amazon.com/images/M/MV5BM2MyNjYxNmUtYTawNi00MTYxLWJmNWYtYzZlODY3ZTk3C  
a schema:Movie ;  
schema:image "https://m.media-  
amazon.com/images/M/MV5BM2MyNjYxNmUtYTawNi00MTYxLWJmNWYtYzZlODY3ZTk3C  
schema:name "The Godfather" ;  
schema:datePublished "1972"^^xsd:gYear ;  
schema:contentRating "A" ;  
schema:duration "PT175M" ;  
schema:genre "Crime, Drama" ;  
schema:aggregateRating [  
a schema:AggregateRating ;  
schema:ratingValue "9.2" ;  
schema:bestRating "10" ;  
schema:worstRating "0" ;  
schema:ratingCount "1620367" ;  
schema:reviewCount "134,966,411"  
];  
schema:description "An organized crime dynasty's aging patriarch transfers control of his cl  
schema:review [  
a schema:Review ;  
schema:reviewRating [  
a schema:Rating ;  
schema:ratingValue "100"  
]  
];  
schema:director [  
a schema:Person ;  
schema:name "Francis Ford Coppola"  
];
```

# RML – Few-Shot

```
turtle Copy code

@prefix rr: <http://www.w3.org/ns/r2rml#> .
@prefix rml: <http://semweb.mmlab.be/ns/rml#> .
@prefix ql: <http://semweb.mmlab.be/ns/ql#> .
@prefix schema: <http://schema.org/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<#FilmMapping>
  a rml:TriplesMap ;
  rml:logicalSource [
    rml:source "path/to/film-dataset.csv" ;
    rml:referenceFormulation ql:CSV
  ] ;
  rr:subjectMap [
    rr:template "https://m.media-amazon.com/images/M/{Image}" ;
    rr:class schema:Movie
  ] ;
  rr:predicateObjectMap [
    rr:predicate schema:image ;
    rr:objectMap [
      rr:column "Image"
    ]
  ] ;
  rr:predicateObjectMap [
    rr:predicate schema:name ;
    rr:objectMap [
      rr:column "Name"
    ]
  ] ;
  rr:predicateObjectMap [
    rr:predicate schema:datePublished ;
    rr:objectMap [
      rr:column "Year" ;
      rr:datatype xsd:gYear
    ]
  ] ;
```

```
rr:predicateObjectMap [
  rr:predicate schema:contentRating ;
  rr:objectMap [
    rr:column "ContentRating"
  ]
] ;
rr:predicateObjectMap [
  rr:predicate schema:duration ;
  rr:objectMap [
    rr:column "Duration"
  ]
] ;
rr:predicateObjectMap [
  rr:predicate schema:genre ;
  rr:objectMap [
    rr:column "Genre"
  ]
] ;
rr:predicateObjectMap [
  rr:predicate schema:description ;
  rr:objectMap [
    rr:column "Description"
  ]
] ;
rr:predicateObjectMap [
  rr:predicate schema:aggregateRating ;
  rr:objectMap [
    rr:template [
      rr:column "Rating"
    ] ;
    rr:termType rr:Literal ;
    rr:datatype xsd:double
  ]
] ;
```

# Comparisons

RDF generation vs RML generation

- RML generates template
- Usable on many instances
- RDF limited by characters
- Both methods failed to immediately apply schema.org



# Comparisons

## Zero-Shot vs One-Shot vs Few-Shot

- All methods eventually produced good results
- Difference in required fine-tuning amount
- Zero-shot already works great when all the info provided immediately
- One-shot did too much specificity
- No alterations required for Few-shot

# Datasets

## CSV file Containing Video games Dataset

Columns: 17

Dataset: 19714

### Dataset:

Id, name title, IMDB URL, Year of release, certificate ratings , average rating, number of votes, summary plot, Action is game of action genre?, Adventure is game of adventure genre?,Comedy, Crime, Family, Fantasy, Mystery, Sci-Fi, Thriller.

# Mapping

- Prompt Engineering :
- Using ChatGPT Version GPT-4

**Convert CVS file dataset into RDF.**

- RML Generation
- One-Shot
- Few-Shot

**Convert CVS file dataset into RDF.**

- RML Generation
- Zero-Shot
- One-Shot
- Few-Shot

# RDF Generation.

- One-shot

SR

You

Convert the following data into RDF

id, name, url year, certificate, rating, votes, plot, Action, Adventure, Comedy, Crime, Family, Fantasy, Mystery, Sci-Fi, Thriller

"1, Red Dead Redemption II, https://www.imdb.com/title/tt6161168/?ref=adv\_li\_tt, 2018, M, 9.7, 35,703, Amidst the decline of the Wild West at the turn of the 20th century, TRUE, TRUE, FALSE, TRUE, FALSE, FALSE, FALSE, FALSE, FALSE"

@prefix ex: <http://example.org/game/> .  
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

ex:1 a ex:VideoGame ;  
ex:id "0" ;  
ex:name "Spider-Man" ;  
ex:url <https://www.imdb.com/title/tt5807780/?ref=adv\_li\_tt> ;  
ex:year "2018" ;  
ex:certificate "M" ;  
ex:rating "9.2" ;  
ex:votes "20,759" ;  
ex:plot "When a new villain threatens New York City" ;  
ex:genre ex:Action, ex:Adventure, ex:Crime ;  
ex:hasFamilyGenre "FALSE" ;  
ex:hasFantasyGenre "TRUE" ;  
ex:hasMysteryGenre "FALSE" ;  
ex:hasSciFiGenre "FALSE" ;  
ex:hasThrillerGenre "FALSE" .

Given CSV data:

```
less Copy code

"1, Red Dead Redemption II, https://www.imdb.com/title/tt6161168/?ref=
```

RDF representation:

```
turtle Copy code

@prefix ex: <http://example.org/game/> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

ex:1 a ex:VideoGame ;
  ex:id "1" ;
  ex:name "Red Dead Redemption II" ;
  ex:url <https://www.imdb.com/title/tt6161168/?ref=adv_li_tt> ;
  ex:year "2018" ;
  ex:certificate "M" ;
  ex:rating "9.7" ;
  ex:votes "35,703" ;
  ex:plot "Amidst the decline of the Wild West at the turn of the 20t
  ex:genre ex:Action, ex:Adventure, ex:Crime ;
  ex:hasFamilyGenre "FALSE" ;
  ex:hasFantasyGenre "FALSE" ;
  ex:hasMysteryGenre "FALSE" ;
  ex:hasSciFiGenre "FALSE" ;
  ex:hasThrillerGenre "FALSE" .
```

# RDF Generation.

- Few-shots
- Give Multiple Input to get the for Completing The request

**SR** You  
convert the provided CSV data for provided RDF structure. Combine all of the three dataset of id '0' id '1' and id '2'

id, name, url, year, certificate, rating, votes, plot, Action, Adventure, Comedy, Crime, Family, Fantasy, Mystery, Sci-Fi, Thriller

"1, Red Dead Redemption II, https://www.imdb.com/title/tt6161168/?ref\_=adv\_li\_tt, 2018, M, 9.7, 35,703, Amidst the decline of the Wild West at the turn of the 20th century, TRUE, TRUE, FALSE, TRUE, FALSE, FALSE, FALSE, FALSE, FALSE"

@prefix ex: <http://example.org/game/> .  
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

```
ex:1 a ex:VideoGame ;
  ex:id "0" ;
  ex:name "Spider-Man" ;
  ex:url <https://www.imdb.com/title/tt5807780/?ref_=adv_li_tt> ;
  ex:year "2018" ;
  ex:certificate "M" ;
  ex:rating "9.2" ;
  ex:votes "20,759" ;
  ex:plot "When a new villain threatens New York City" ;
  ex:genre ex:Action, ex:Adventure, ex:Crime ;
  ex:hasFamilyGenre "FALSE" ;
  ex:hasFantasyGenre "TRUE" ;
  ex:hasMysteryGenre "FALSE" ;
  ex:hasSciFiGenre "FALSE" ;
  ex:hasThrillerGenre "FALSE" .

###
```

id, name, url, year, certificate, rating, votes, plot, Action, Adventure, Comedy, Crime, Family, Fantasy, Mystery, Sci-Fi, Thriller

"0, Spider-Man, https://www.imdb.com/title/tt5807780/?ref\_=adv\_li\_tt, 2018, T, 9.2, 20,759, When a new villain threatens New York City, TRUE, TRUE, FALSE, FALSE, FALSE, TRUE, FALSE, FALSE, FALSE, FALSE"

id '2' is: "2, Grand Theft Auto V, https://www.imdb.com/title/tt2103188/?ref\_=adv\_li\_tt, 2013, M, 9.5, 59,986, Three very different criminals team up for a series of heists and walk into some of the most thrilling experiences in the corrupt city of Los Santos., TRUE, FALSE, FALSE, TRUE, FALSE, FALSE, FALSE, FALSE, FALSE"

```
turtle Copy code

@prefix ex: <http://example.org/game/> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

ex:0 a ex:VideoGame ;
  ex:id "0" ;
  ex:name "Spider-Man" ;
  ex:url <https://www.imdb.com/title/tt5807780/?ref_=adv_li_tt> ;
  ex:year "2018" ;
  ex:certificate "T" ;
  ex:rating "9.2" ;
  ex:votes "20,759" ;
  ex:plot "When a new villain threatens New York City" ;
  ex:genre ex:Action, ex:Adventure, ex:Fantasy ;
  ex:hasFamilyGenre "FALSE" ;
  ex:hasMysteryGenre "FALSE" ;
  ex:hasSciFiGenre "FALSE" ;
  ex:hasThrillerGenre "FALSE" .

ex:1 a ex:VideoGame ;
  ex:id "1" ;
  ex:name "Red Dead Redemption II" ;
  ex:url <https://www.imdb.com/title/tt6161168/?ref_=adv_li_tt> ;
  ex:year "2018" ;
  ex:certificate "M" ;
  ex:rating "9.7" ;
  ex:votes "35,703" ;
  ex:plot "Amidst the decline of the Wild West at the turn of the 20th century" ;
  ex:genre ex:Action, ex:Adventure, ex:Crime ;
  ex:hasFamilyGenre "FALSE" ;
  ex:hasFantasyGenre "FALSE" ;
  ex:hasMysteryGenre "FALSE" ;
  ex:hasSciFiGenre "FALSE" ;
  ex:hasThrillerGenre "FALSE" .

ex:2 a ex:VideoGame ;
  ex:id "2" ;
  ex:name "Grand Theft Auto V" ;
  ex:url <https://www.imdb.com/title/tt2103188/?ref_=adv_li_tt> ;
  ex:year "2013" ;
  ex:certificate "M" ;
  ex:rating "9.5" ;
```

# Mapping

- Prompt Engineering :
- Using ChatGPT Version GPT-4

**Convert CVS file dataset into RML.**

- RML Generation
- Zero-Shot
- One-Shot
- Few-Shot

# RML Generation.

- Zero-shot
  - No previous information and data is provided for the completing request.

SR

You



imdb-videogames.csv  
Spreadsheet

Generate RML file to convert it to RDF.

```
turtle
Copy code

@prefix rr: <http://www.w3.org/ns/r2rml#>.
@prefix rml: <http://semweb.mmlab.be/ns/rml#>.
@prefix ql: <http://semweb.mmlab.be/ns/ql#>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
@prefix ex: <http://example.com/ns#>.

<#VideoGameMapping>
  rml:logicalSource [
    rml:source "/path/to/imdb-videogames.csv";
    rml:referenceFormulation ql:CSV
  ];
  rr:subjectMap [
    rr:template "http://example.com/videogame/{id}";
    rr:class ex:VideoGame
  ];
  rr:predicateObjectMap [
    rr:predicate ex:name;
    rr:objectMap [ rr:column "name" ]
  ];
  rr:predicateObjectMap [
    rr:predicate ex:url;
    rr:objectMap [ rr:column "url" ]
  ];
  rr:predicateObjectMap [
    rr:predicate ex:year;
    rr:objectMap [ rr:column "year"; rr:datatype xsd:integer ]
  ];
  rr:predicateObjectMap [
    rr:predicate ex:certificate;
    rr:objectMap [ rr:column "certificate" ]
  ];
  rr:predicateObjectMap [
    rr:predicate ex:rating;
    rr:objectMap [ rr:column "rating"; rr:datatype xsd:decimal ]
  ];
  rr:predicateObjectMap [
    rr:predicate ex:votes;
    rr:objectMap [ rr:column "votes" ]
  ];
  rr:predicateObjectMap [
```

```
rr:predicateObjectMap [
  rr:predicate ex:plot;
  rr:objectMap [ rr:column "plot" ]
];
# Mapping for genres (e.g., Action, Adventure, etc.)
rr:predicateObjectMap [
  rr:predicate ex:genre;
  rr:objectMap [
    rr:column "Action";
    rr:datatype xsd:boolean;
    rr:termType rr:Literal
  ]
];
```



# RML Generation.

- Zero-shot
- Applying Scheme.org Ontologies for rml

Now, let's adjust the RML mapping to use these Schema.org properties. I'll modify the previous RML template to reflect these changes.

```
turtle
Copy code

@prefix rr: <http://www.w3.org/ns/r2rml#>.
@prefix rml: <http://semweb.mmlab.be/ns/rml#>.
@prefix ql: <http://semweb.mmlab.be/ns/ql#>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
@prefix schema: <http://schema.org/>.

<#VideoGameMapping>
  rml:logicalSource [
    rml:source "/path/to/imdb-videogames.csv";
    rml:referenceFormulation ql:CSV
  ];
  rr:subjectMap [
    rr:template "http://example.com/videogame/{id}";
    rr:class schema:VideoGame
  ];
  rr:predicateObjectMap [
    rr:predicate schema:name;
    rr:objectMap [ rr:column "name" ]
  ];
  rr:predicateObjectMap [
    rr:predicate schema:url;
    rr:objectMap [ rr:column "url" ]
  ];
  rr:predicateObjectMap [
    rr:predicate schema:datePublished;
    rr:objectMap [ rr:column "year"; rr:datatype xsd:gYear ]
  ];
  rr:predicateObjectMap [
    rr:predicate schema:contentRating;
    rr:objectMap [ rr:column "certificate" ]
  ];
  rr:predicateObjectMap [
    rr:predicate schema:aggregateRating;
    rr:objectMap [
      rr:parentTriplesMap <#RatingMapping>
    ]
  ];
  rr:predicateObjectMap [
```

```
rr:objectMap [ rr:column "rating"; rr:datatype xsd:decimal ]
];
rr:predicateObjectMap [
  rr:predicate schema:reviewCount;
  rr:objectMap [ rr:column "votes" ]
];
```

n this revised RML mapping:



You

As you mentioned the column of CSV file are as follows.

id: A unique identifier for each video game.

name: The name of the video game.

url: A URL to the video game's page on IMDb.

year: The release year of the video game.

certificate: The game's rating certificate.

rating: The IMDb rating of the video game.

votes: The number of votes the game received on IMDb.

plot: A brief plot description of the video game.

Action, Adventure, Comedy, Crime, Family, Fantasy, Mystery, Sci-Fi, Thriller: Boolean

columns indicating the genres of the video game.

Apply sheme.org ontologies to them.

# RML Generation.

- One-shot
- One Specific Input is provided and given in [scheme.org](http://scheme.org) ontologies.



You  
convert this Data into RML format.

"1, Red Dead Redemption II, [https://www.imdb.com/title/tt6161168/?ref\\_=adv\\_li\\_tt](https://www.imdb.com/title/tt6161168/?ref_=adv_li_tt),  
2018, M, 9.7, 35703, Amidst the decline of the Wild West at the turn of the 20th  
century, TRUE, TRUE, FALSE, TRUE, FALSE, FALSE, FALSE, FALSE, FALSE"

turtle

Copy code

```
<http://example.com/videogame/1> a schema:VideoGame;  
    schema:name "Red Dead Redemption II";  
    schema:url <https://www.imdb.com/title/tt6161168/?ref_=adv_li_tt>;  
    schema:datePublished "2018"^^xsd:gYear;  
    schema:contentRating "M";  
    schema:description "Amidst the decline of the Wild West at the turn  
    schema:aggregateRating [  
        schema:ratingValue "9.7"^^xsd:decimal;  
        schema:reviewCount "35703"^^xsd:integer;  
    ];  
    schema:genre "Action", "Adventure", "Crime".
```

# RML Generation.

- Few-shots
- Multiple pieces of data or guidelines given before completing request
- Generate Mapping for first two Unique ID: id '0' and id '1'



You

Generate RML mapping for these two id '0' and id '0'

"0, Spider-Man, [https://www.imdb.com/title/tt5807780/?ref\\_=adv\\_li\\_tt](https://www.imdb.com/title/tt5807780/?ref_=adv_li_tt), 2018, T, 9.2, 20,759, When a new villain threatens New York City, TRUE, TRUE, FALSE, FALSE, FALSE, TRUE, FALSE, FALSE, FALSE"

id '2' is: "2, Grand Theft Auto V, [https://www.imdb.com/title/tt2103188/?ref\\_=adv\\_li\\_tt](https://www.imdb.com/title/tt2103188/?ref_=adv_li_tt), 2013, M, 9.5, 59,986, Three very different criminals team up for a series of heists and walk into some of the most thrilling experiences in the corrupt city of Los Santos., TRUE, FALSE, FALSE, TRUE, FALSE, FALSE, FALSE, FALSE, FALSE"

# RML Generation.

- Few-shots

```
turtle Copy code

@prefix rr: <http://www.w3.org/ns/r2rml#>.
@prefix rml: <http://semweb.mmlab.be/ns/rml#>.
@prefix ql: <http://semweb.mmlab.be/ns/ql#>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
@prefix ex: <http://example.org/game/>.

<#VideoGameMapping>
  rml:logicalSource [
    rml:source "/path/to/your/csvfile.csv";
    rml:referenceFormulation ql:CSV;
    rml:iterator "row"
  ];
  rr:subjectMap [
    rr:template "http://example.org/game/{id}";
    rr:class ex:VideoGame
  ];
  rr:predicateObjectMap [
    rr:predicate ex:id;
    rr:objectMap [ rml:reference "id" ]
  ];
  rr:predicateObjectMap [
    rr:predicate ex:name;
    rr:objectMap [ rml:reference "name" ]
  ];
  rr:predicateObjectMap [
    rr:predicate ex:url;
    rr:objectMap [ rml:reference "url" ]
  ];
  rr:predicateObjectMap [
    rr:predicate ex:year;
    rr:objectMap [ rml:reference "year"; rr:datatype xsd:gYear ]
  ];
  rr:predicateObjectMap [
    rr:predicate ex:certificate;
    rr:objectMap [ rml:reference "certificate" ]
  ];
  rr:predicateObjectMap [
    rr:predicate ex:rating;
    rr:objectMap [ rml:reference "rating"; rr:datatype xsd:decimal
  ];
  rr:predicateObjectMap [
```

```
rr:predicateObjectMap [
  rr:predicate ex:plot;
  rr:objectMap [ rml:reference "plot" ]
];
# Genre mappings for Action, Adventure, etc.
rr:predicateObjectMap [
  rr:predicate ex:genre;
  rr:objectMap [
    rml:reference "Action";
    rr:datatype xsd:boolean
  ]
];
```

# Conclusions

- Prompt engineering offers great promise
- Great results for both producing RDF and RML
- No schema ontologies as it wants to remain as broad as possible
- Amount of information given directly correlates to result
- Great way of skipping manual RML creation
- More usable than existing tools (YARRRML)



