

# **Gamification to enhance Wikidata for Indian Languages**

## **Architecture Document**

Group No: 13 (Eesha Dutta, Paryul Jain, Kumar Pallav, Amit Sarkar)

Mentor: Himanshu Maheshwari

---

### **Abstract**

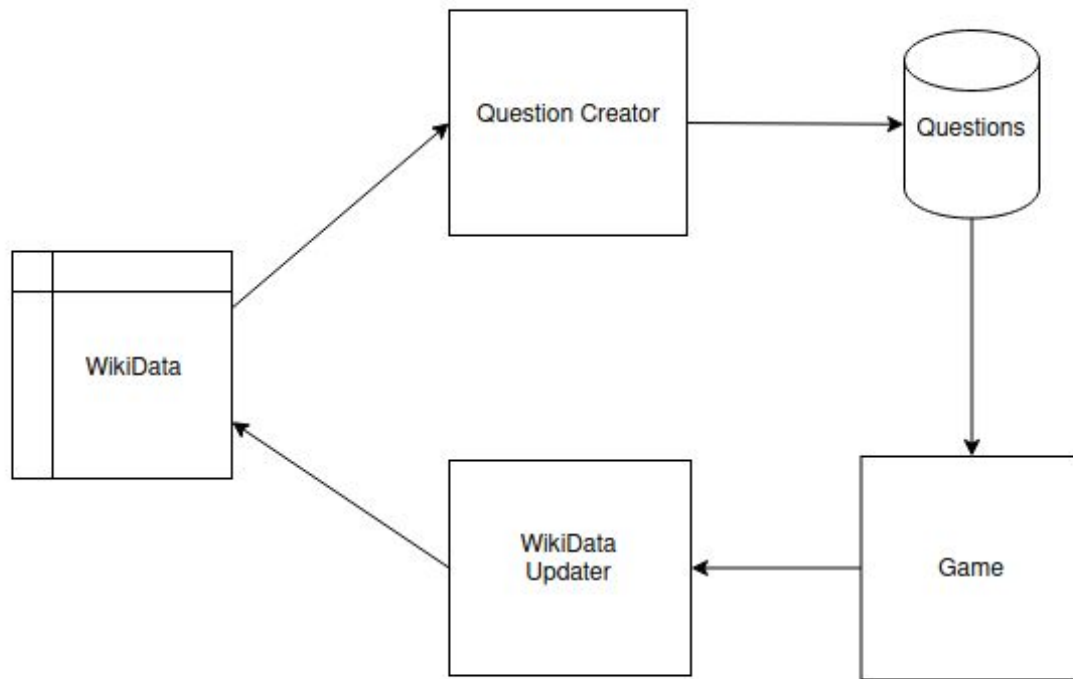
Wikidata is a free structured knowledge base that can be read and edited by both humans and machines. Wikipedia is a multilingual, web-based, free-content encyclopedia project supported by the Wikimedia Foundation and based on a model of openly editable content. Wikidata has the potential to provide factual data that could be leveraged by the Wikipedia writers.

Gamification is the application of typical elements of game playing (e.g. point scoring, competition with others, rules of play) to other areas of activity (here enhancing Wikidata for Hindi and Telugu), to encourage engagement with a product or service. This survey talks about a few games that aim to enhance English wiki data viz. The Wikidata Game, Wikidata Gameabc, etc.

### **Methodologies, Architecture, and WorkFlow**

We started by dividing the project into 3 modules. Question Creator, The Game, and Wiki Updater. The question creator was focused on extracting fields which could be filled, and accordingly fed to the game engine where we can get accurate answers by checking the provided references using IR techniques. Once we reach a confidence value in a question we then update it's answer onto the wikitable, and remove the question from contention among users.

## Overall Architecture



## Question Creator

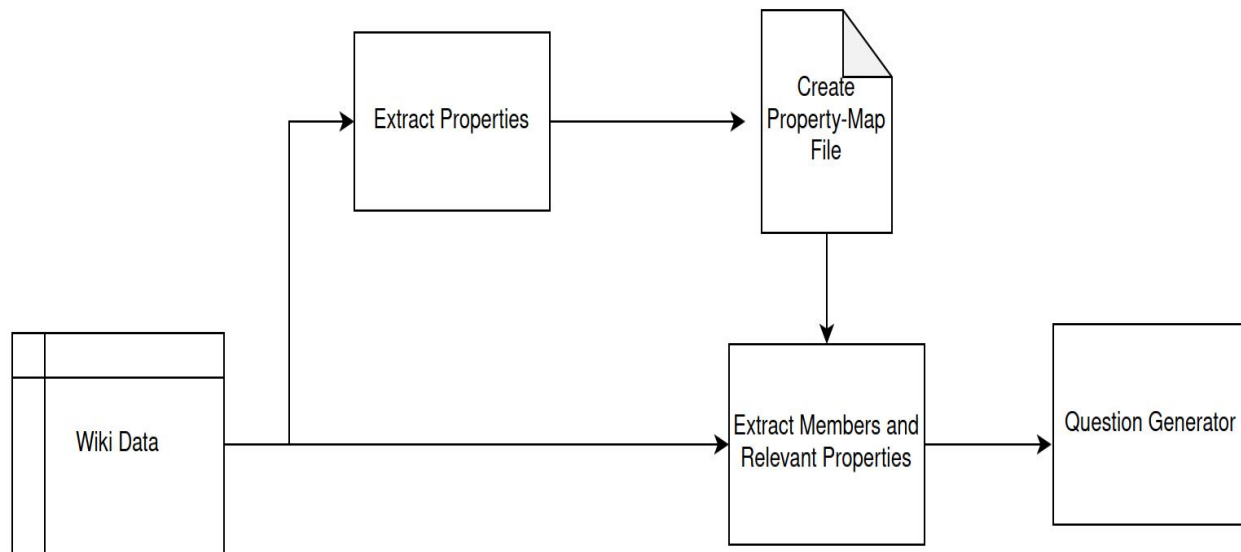


Fig: Question Creator flow-diagram

### Question Creator Flow:

- 1.Extract Properties:** this module will extract all possible properties of a given instance. We store all properties of all elements belonging to a particular category (say chemical elements).
- 2. Create a set of properties missing for elements:** In this part, we take two elements and create a difference set (properties missing in an element).
- 3. Validate missing properties:** In this section, we consider which of the missing properties belong to at least 75 percent of elements as there might be a property specific to an element.
- 4. Create questions:** In this part, the questions are created by using a template where we place the elements name and its missing property to generate the question.

## Game Engine - Database Schemas

### USER

Fullname (STRING)

Username (VARCHAR) (UNIQUE)

Password (VARCHAR) (HASHED)

Image (IMAGE)

Trustscore (INTEGER)

Last test score (INTEGER)

### QUESTION

Question ID-OVR (INTEGER)

Genre name (STRING)

Question ID-gen (INTEGER)

Question-Q (STRING)

Answer (STRING)

Reference (STRING)

Question-t (STRING) (OPTIONAL)

Correctness-score (INTEGER)

### ANSWER

Question ID-OVR (INTEGER)

Answer-1 (STRING)

Answer-2 (STRING)

Confidence-score1 (~~INT~~)

Confidence-score2 (~~INT~~)

Confidence-score-OVR (INTEGER)

## Game Workflow

The users will be asked to login/signup and will be directed to the main dashboard, where if they have not already can update their profile picture.

The users will be shown options for multiplayer and single player, from which they can choose which mode to compete. (Currently we are focused on single player and the gameflow is described for single player.)

After choosing the mode, the player will be displayed several genres in which they want to play. Once the genre is selected, the player will be given instructions on how to fill the answer within the time limit. This page will vary for different modes of play, different genres and different types of questions.

Once the genre is chosen and the game is started, the user will be displayed some questions, to which he has to give an answer along with a reference, or simply give an answer or just skip the answer. References, if provided will be used to check the correctness of the answer.

The different modes will be scored accordingly depending upon the correctness.

Answer already Existing	Reference Checker	Trust Score(User)	Confidence Score(Answer)
No	Yes/No/N.P(Not Provided)	0.5/-0.1/0.05	0.1//
Yes	Yes/No/N.P	0.3/0.01/0.1	0.2//

The scores are assigned to the user and the question so that a trusted user has more weightage at the end, while updating the answer. Similarly once the confidence score of an answer to a specific question reaches above a threshold, then we can drop the question from getting displayed to the users and it will finally be updated in the wiki table.

## Reference Checker

Once we get the answer from a user, and it's the corresponding reference, we then need to verify the credibility of the source and the authenticity of the user and assign scores accordingly.

A very naive approach to check the correctness would be to use the IR engine developed during the mini-project. We would be sending out the missing attribute, the value provided by the user, and the object of which the attribute is in question.

This problem can be solved better if you also include the background text from where the questions have been picked up. Then, firstly train word embeddings on the background text. Further, generate sentence compositionality of the questions as well as their answers using a Recursive Neural Network or some variant. After all these steps you can compute the similarity between compositions of the new answer and given answers to know if it is correct or not. This kind of question-answer verification will not only be restricted to accurately checking facts only question but also more open-ended questions since they will be comparing the meaning of the two texts. This has been implemented using the Siamese network, training the embeddings, and then uses LSTM and it's variants, coupled with co-sine similarity and other similarity measures.

## Wiki Table Entry

The final wiki table entry structure is in process. We read about the bot which enters the data, but will give a detailed description in the next iteration.

## Code and Documents

Code Link for Game Engine: <https://github.com/eeshadutta/Gamification-to-Enhance-Wikidata>

Code Link for Question Answering and Similarity checking in references:

[https://github.com/Paryul10/Community-Question\\_Answering](https://github.com/Paryul10/Community-Question_Answering)

Code Link for Minor-Project (IRE Engine):

<https://github.com/Paryul10/Wikipedia-Search-Engine>

## Upcoming Advancements and Difficulties Till Now

Currently we are in the phase/almost done to implement the very basic approaches so that we can have a final End-to-End system ready where we can plugin or remove or modify any module as and when we make advancements. We would be exploring techniques similar to Semantic Similarity one described above for reference checking and validation of answers (DrQA, RankQA). Also, a fusion of Search and Retrieval coupled with Semantic matching is expected to give better results overall.

We would also be exploring rule-based approaches for question generation .... (@AmitSarkar and @PallavKumar)