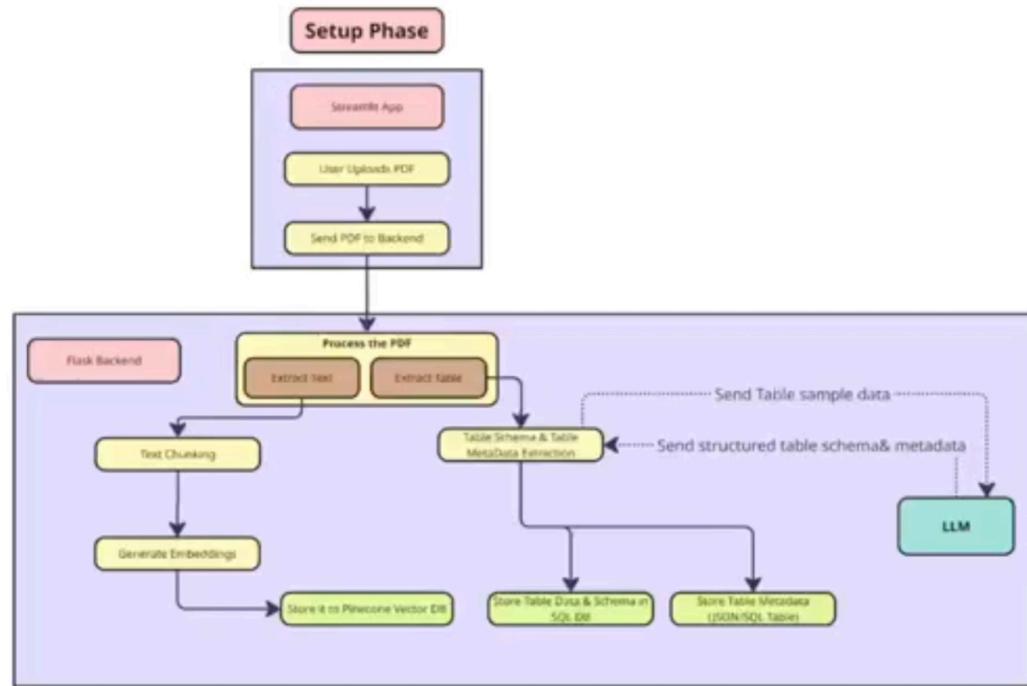


# HybridRAG



**Suggested Title:** "Hybrid RAG: Solving the Problem of Tables in PDFs with Digitized Data"

So, let us illustrate this concept with the help of a PDF file which consists of FIFA World Cup data, correct.

So, this PDF consists of text which you can see which consists of the various details of World Cup and also the data of the World Cup match results in the form of a table, ok.

So, you can see that it contains the data of each year and what are the various rounds and who won the finals by board score ok.

So, 1930 is 1 final, 1934 is another final, 1938 is another final and so on so forth.

So, this particular PDF has all this data and the problem with a conventional rag is that this particular table gets flattened when we create it into chunks and create

vector embeddings.

Correct.

So if you ask the query on this particular table, it may not be able to give accurate results.

Now, you don't have to believe me, I'm going to show you how it works.

So let's say we have this application which we have built, it's called the Hybrid RAG.

In this, we have fed this particular PDF as the data source and when we ask a particular question right let's pick up a question here how many matches in the World Cup ended in a draw okay now when I ask this particular question to the hybrid rag you will see that it returns a response which is an exact count from this particular PDF, right?

Now, you see hybrid DAG has given an exact result 14, whereas conventional DAG has given all over the place some kind of data.

Now, how is it come, right?

You can see there each match where it is a draw right, you can see it is a draw here, there is 1 more draw here 1938 and so on so forth right.

How is it getting the information of exactly 14 right?

So, what is happening under the hood is the data is getting digitized right, All that table in this particular PDF is converted dynamically using an LLM.

A schema is created and the data is also inserted into super base which is a post SQL database and it is converted into a digital form like this.

Now when we ask a question saying how many matches in the World Cup and if in a draw it is a matter of a simple SQL query which is dynamically constructed from the query from the message that I have asked correct.

So it is a matter of just running this and it gives you the exact 14 rows.

Here I have not asked for what data they are, but it can also give you what exactly are those years where this happened, correct?

So in fact, if you want, we can actually try it out as well, okay?

I will leave it to you to for trying it.

Now, How does this work under the hood, right?

See, under the hood, what we are trying to do is we are taking the data from the PDF and converting, extracting the text and the table separately from the PDF.

And whatever text we get, we're chunking it, creating embeddings and storing it in vector database, which is Pinecone, extracting the table data, creating and generating the schema, taking out the data from the table and storing it in

Superbase, correct?

Now, when the user asks a query, we use agent-to-query architecture to dynamically decide, is it a text query or a table query, and route it to the appropriate agent to fetch the data and return back to the user.

So, let us look at it from a purely text query.

What was the host nation for the first World Cup?

Now, if you look at this, this particular information is not going to come from a table, correct?

What was the first host nation, right?

Because this has the results of the matches which happened.

It doesn't mention about which, what was the host, which was organizing that particular World Cup.

Now you see that data is over here in the text, right?

Uruguay was chosen as the first host nation.

Now, when we ask this particular question to the hybrid RAG, it gives us that accurate answer.

So how is this coming?

It's coming basically because it is a pure text-based query and it is getting the data from there.

So that is the power of hybrid RAG.

If you want, we can also try out 1 of another interesting use case, which is what was the highest home score by any team?

So in this query, we are asking what was the highest home score by any team in World Cup matches?

Print the year, home team name, away team name, winning team name, number of goals scored.

Now you can see clearly the difference, right?

Conventional drag just blurts out some information and says you figure out what you want, right?

Whereas hybrid drag gives an exact quantified information Because all the data is simply coming from this particular database.

And that's the power of hybrid drag.

That's the enhanced power you get out of using this architecture.

See you can clearly see it says 1954 Hungary versus South Korea with 9 goals scored.

So the question was highest home score.

So let's verify this data.

So I go to this table, I go to 1954 and I can see that Hungary versus South Korea with 9 goals is the highest home score in this entire table.

So which is so highly accurate.

So if you have a use case where you really want to be able to query on the text or on the table or hybrid, this is the solution you need.

And this is 1 of the coolest, 1 of the cool applications we've built and I thought I'd share the details with you.

Thank you.