**What is a Knowledge Graph?**

A Knowledge Graph is a structure that portrays information as relationships and concepts/entities connected to each other.

These entities connect through each other through various relationships. These entities are depicted as Nodes and the edges demonstrate the relationship between any 2 entities. KG uses NLP to construct a view of nodes and edges through a process called Semantic Enrichment.

The popularity of the term is strictly **connected with the launch of the** [**Google Knowledge Graph**](https://wordlift.io/blog/en/entity/google-knowledge-graph/) **in 2012** and by the introduction of other large databases by major tech companies, such as Yahoo, Microsoft, AirBnB and Facebook, that have created their own “knowledge graphs” to power semantic searches and enable smarter processing of data.

A knowledge graph **doesn’t speak any particular language**. Language is human; a **knowledge graph gets expressed in open linked data, which is the language of machines.**

**Imagine the entire website built upon a large knowledge graph** made of all the metadata that describes the thing that you write about. That knowledge graph becomes part of a larger graph that comprises the new web. That is the power of the **Semantic Web.**

**Why Should We Use Knowledge Graphs?**

The main reason [all businesses should pay close attention to knowledge graphs](https://wordlift.io/blog/en/why-knowledge-graph/) is their role in [**SEO**](https://wordlift.io/blog/en/entity/search-engine-optimization/)**(Search Engine Optimization)**. Adding structured data to your web pages is a sure [way of increasing your rankings and getting more organic traffic.](https://thenextweb.com/news/what-is-a-knowledge-graph-and-how-does-one-work)

Large scale knowledge graphs are usually known for their ability to support NLP applications like semantic search or dialogue generation.Knowledge graphs are intrinsically incomplete and adopting reinforcement learning gives the model a better chance in targeted searches.

But this approach, too, comes with its own set of challenges, such as an agent arriving at a correct answer whose link to its source is missing from the training graph. The agent can also take up unwanted or incorrect pathways just because there was a correct answer in the past.

**Example :- A Recipe Bot**

<https://www.freecodecamp.org/news/have-you-had-the-talk-with-your-chatbot-about-graph-data-structures-3aaf5c3ae52c/>