Wikidata Embedding Project

Glossary

With this glossary, Wikimedia Deutschland provides an overview of the most important terms related to the Wikidata Embedding Project.

Knowledge Graph
Wikidata is an open knowledge graph. Information about people, places, things, events, concepts, and much more is not only stored in a structured way but also connected and put into relation. Example: “Berlin – capital – Germany.”

Structured Data
Structured data is stored according to clearly defined rules in a fixed format, so that computer programs can easily understand it.
Examples: Wikidata or a table with predefined categories for customer addresses.

By contrast, unstructured data does not follow such clear rules. Computer programs must first analyze it before processing.
Examples: Wikipedia articles — lots of words and long texts without a strict rule system. Photos, videos, or chat histories also fall into this category.

Embedding
In the embedding process, information (e.g., words, sentences, images) is translated into a numerical form so that machine learning models can process it. These numbers form a vector (an ordered list of numbers) that captures the meaning of the information. Example: The word “dog” gets a vector very similar to “puppy” but very different from “car.” This way, programs can recognize which contents are similar, even when the exact same words are not used.

Vector
A vector is the result of the embedding process. Each piece of data receives multiple coordinates, similar to a map — except this “map” represents the semantic space of data. The closer two vectors are, the more similar their meanings. All vectors are stored in a vector database, which can be searched for similar vectors.

Semantic Search
When a program can compare vectors, a new type of search becomes possible: semantic search. Instead of looking for exact words, it looks for meaning.
Example: If someone searches for “vehicle”, the system connects it with “car”, even if the word “car” doesn’t appear.

Semantic search makes it possible to perform queries in natural language, based on vectorized data.

Retrieval-Augmented Generation (RAG)
Sometimes the datasets used to train a generative AI are incomplete or outdated. RAG solves this problem by enabling the language model to retrieve additional facts before it generates an answer. This way, it can produce results based on up-to-date knowledge.

Reranker
When a search delivers many results, the first one isn’t always the most relevant. A reranker is a specialized large language model that reorders the results according to relevance.