

Graphs are cool

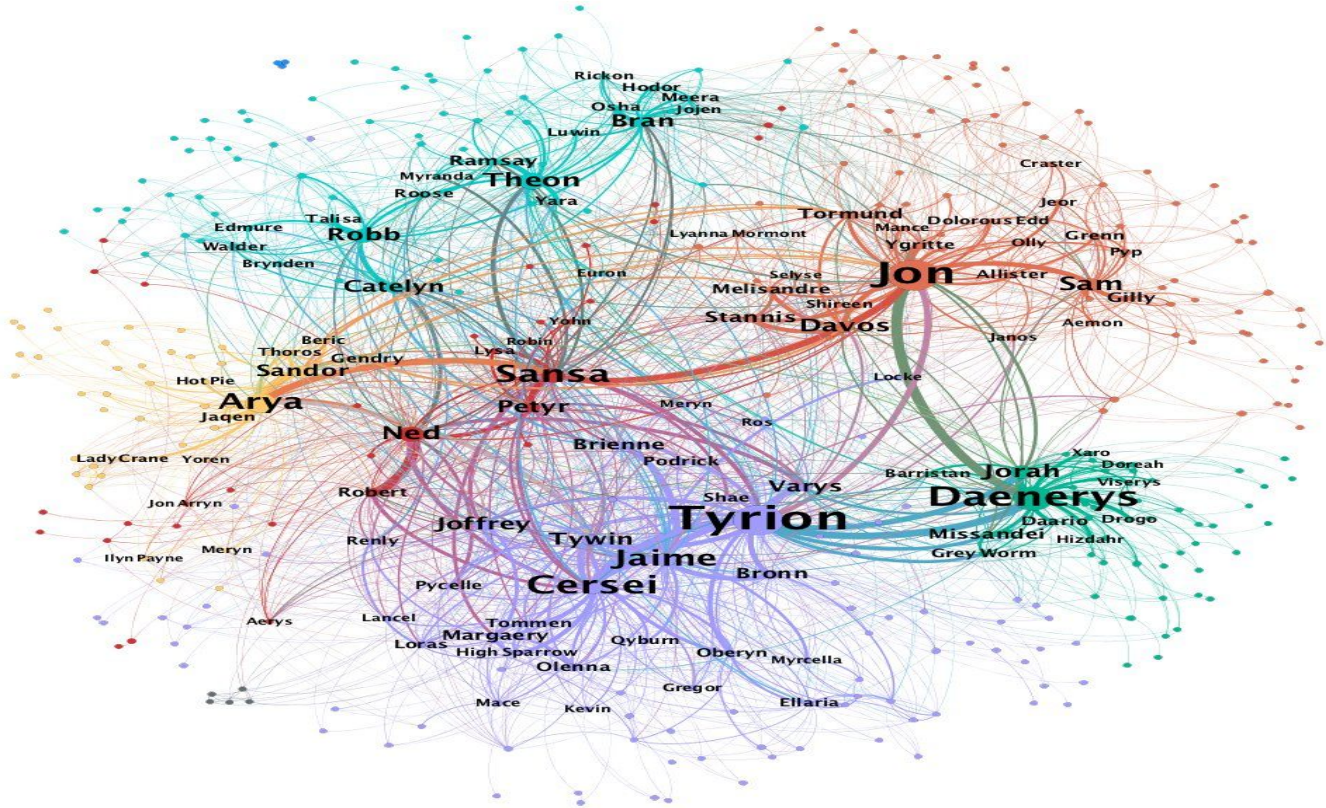
By Tomaž Bratanič



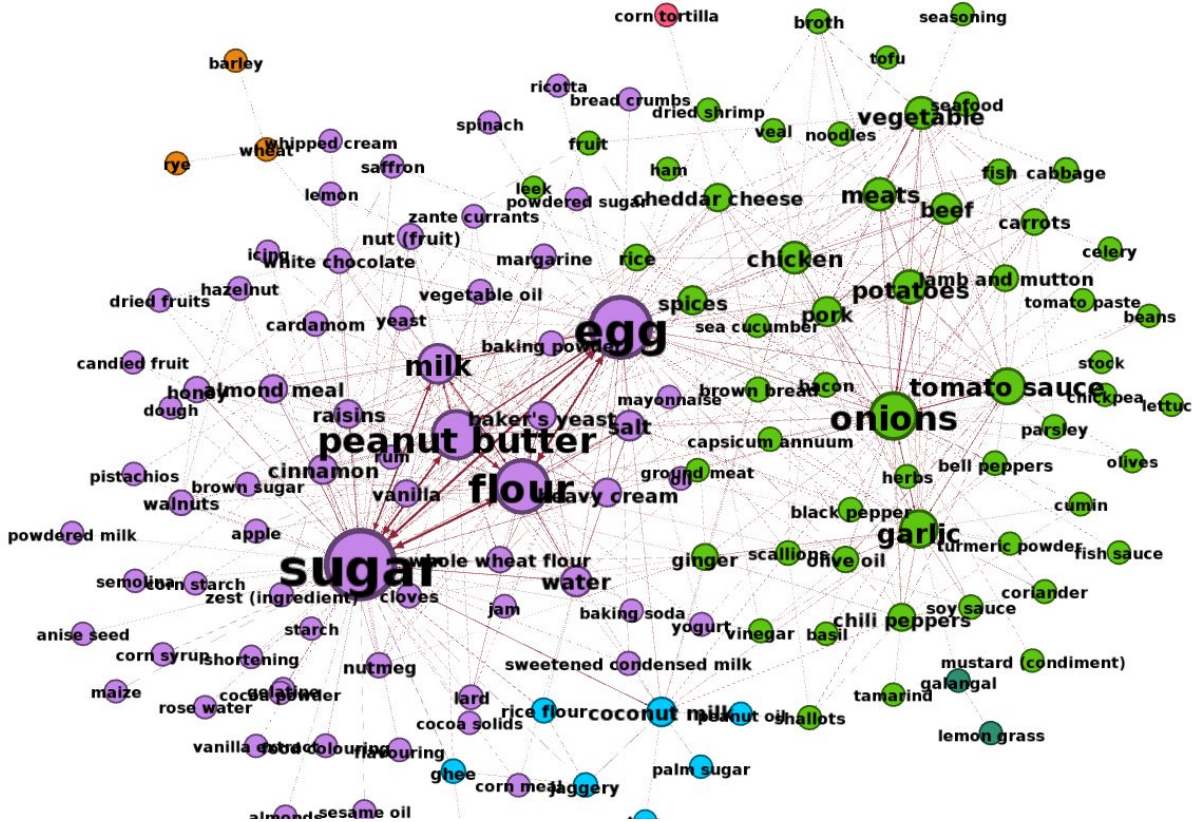
Social networks



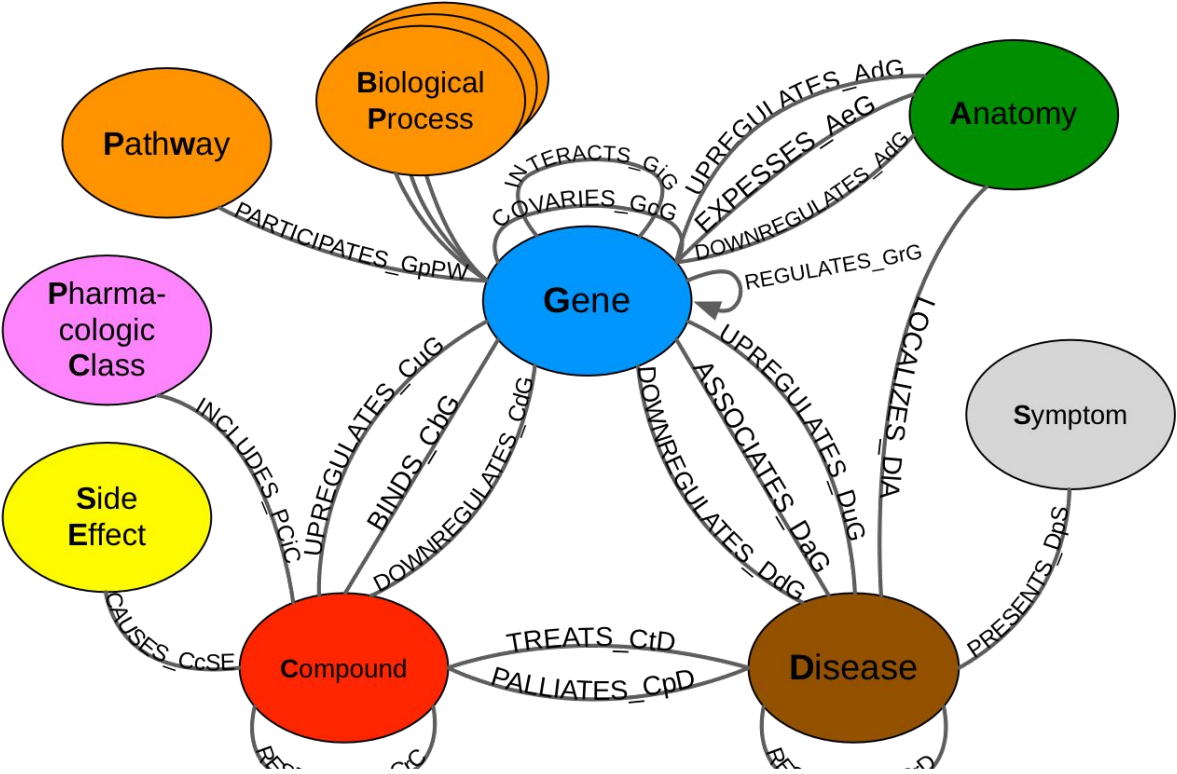
Game of Thrones



Recipes



Biomedical graphs

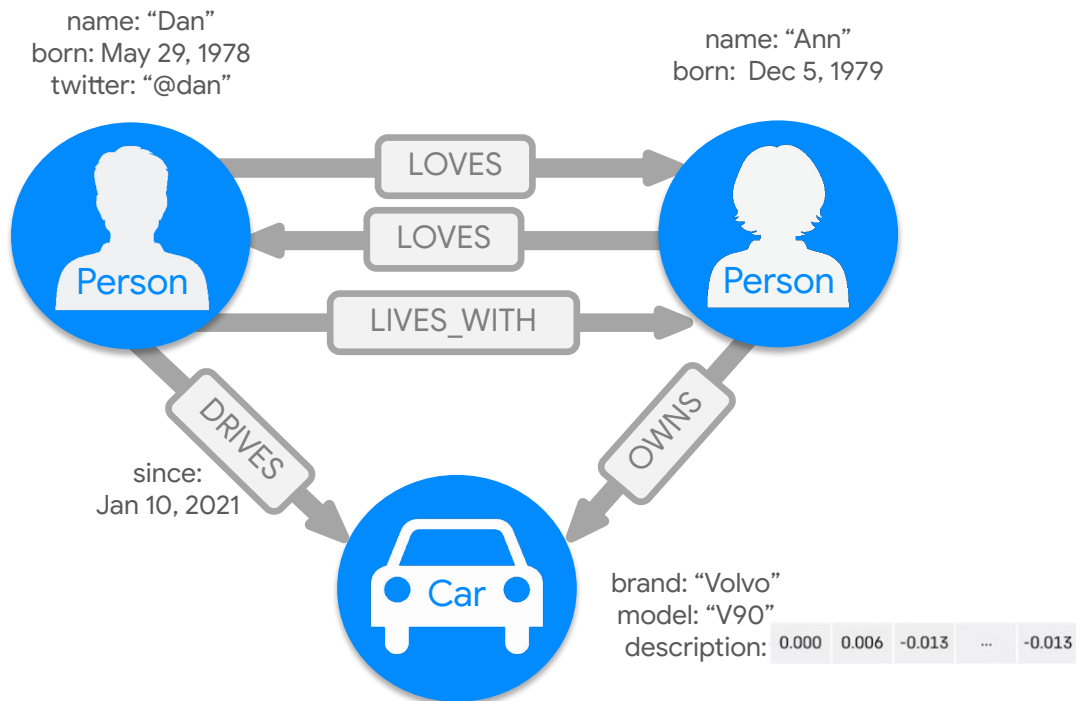


What are Knowledge Graphs?

A knowledge graph is a structured representation of facts, consisting of entities, relationships and semantic descriptions

- Entities can be real-world objects and abstract concepts
- Relationships represent the connections between entities
- Semantic description of the entities and relationships

From Data points to a Knowledge Graph



65+ Graph Algorithms - Out of the Box



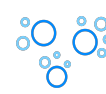
Pathfinding & Search

- Delta-Stepping Single-Source
- Dijkstra's Single-Source
- Dijkstra Source-Target
- All Pairs Shortest Path
- A* Shortest Path
- Yen's K Shortest Path
- Minimum Weight Spanning Tree**
- Random Walk
- Breadth & Depth First Search



Centrality

- Degree Centrality
- Closeness Centrality
- Harmonic Centrality
- Betweenness Centrality** & Approx.
- PageRank
- Personalized PageRank
- ArticleRank
- Eigenvector Centrality
- Hyperlink Induced Topic Search (HITS)
- Influence Maximization (Greedy, CELF)



Community Detection

- Weakly Connected Components**
- Strongly Connected Components
- Label Propagation
- Leiden
- Louvain
- K-Means Clustering
- K-1 Coloring
- Modularity Optimization
- Speaker Listener Label Propagation
- Approximate Max K-Cut
- Triangle Count
- Local Clustering Coefficient
- Conductance Metric



Heuristic LP

- Adamic Adar
- Common Neighbors
- Preferential Attachment
- Resource Allocations
- Same Community
- Total Neighbors



Similarity

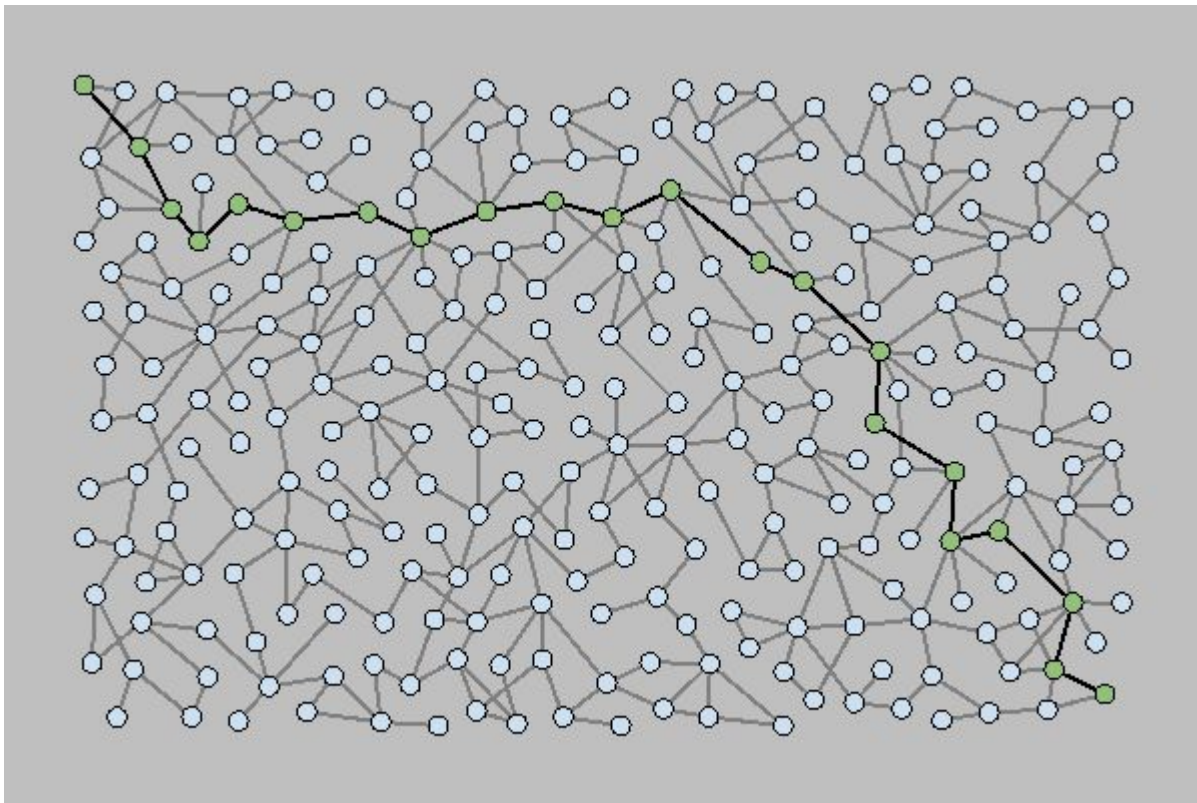
- K-Nearest Neighbors (KNN)
- Filtered K-Nearest Neighbors (KNN)
- Node Similarity**
- Filtered Node Similarity
- Similarity Functions



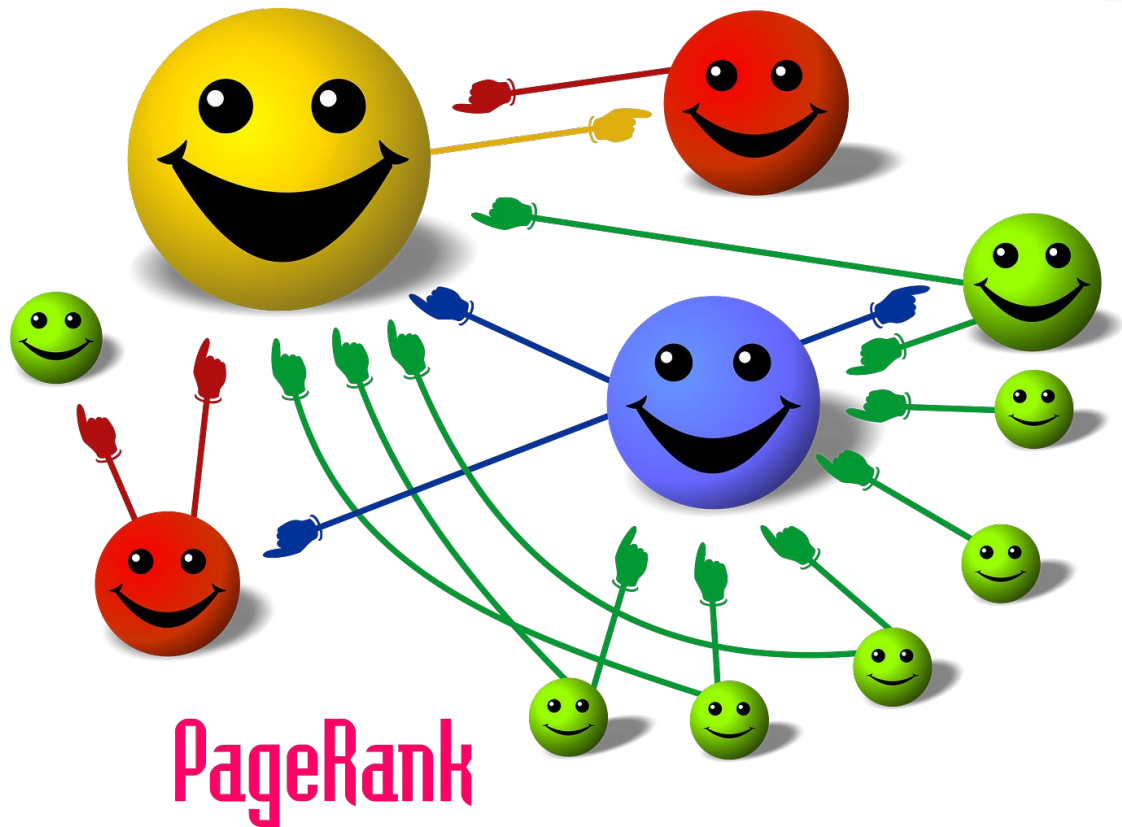
Graph Embeddings

- Fast Random Projection (FastRP)**
- Node2Vec
- GraphSAGE
- HashGNN

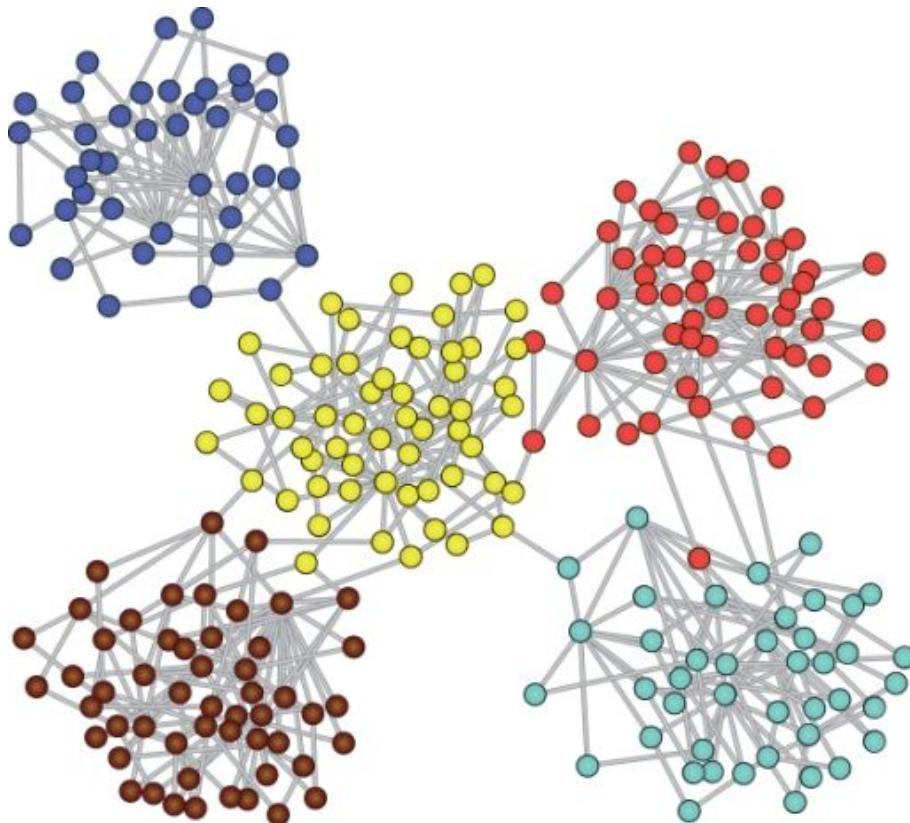
Pathfinding - finding the shortest path



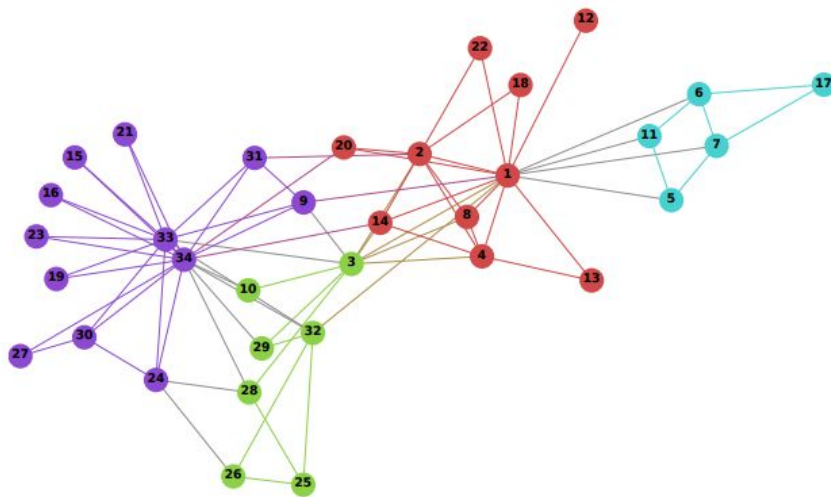
Centrality - finding the most important nodes



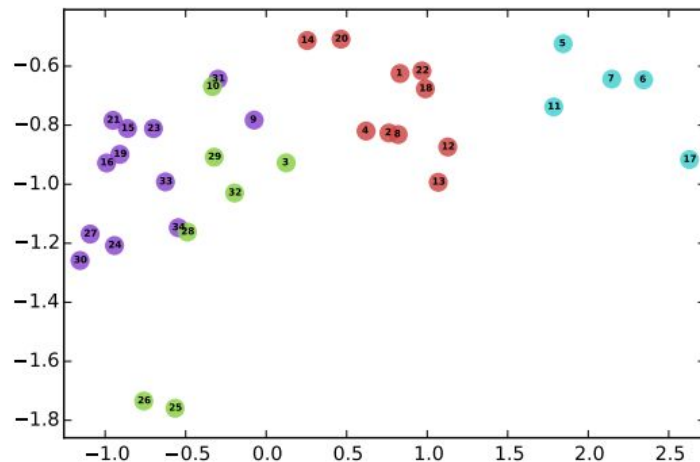
Community detection - finding densely connected groups



Node embeddings - for ML workflows

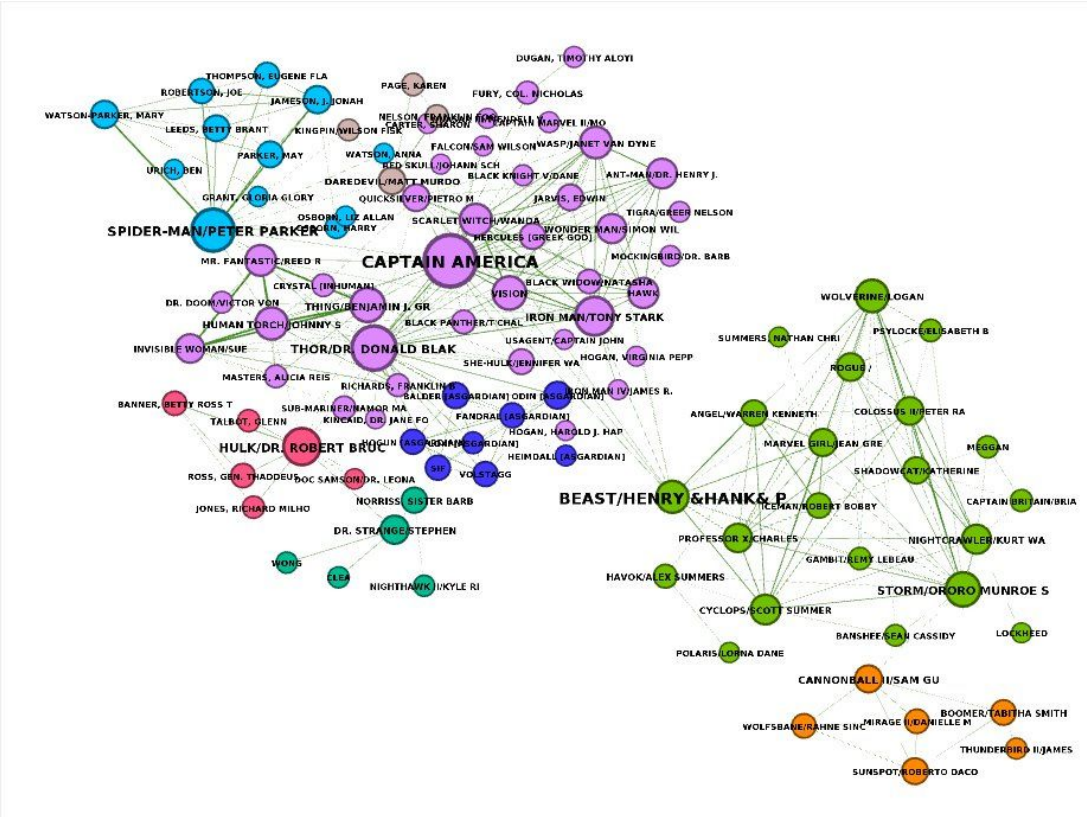


(a) Input: Karate Graph



(b) Output: Representation

Demo time - Marvel



Knowledge-graph-based chatbot



LLMs are great, but:

- Knowledge cutoff date
- Hallucinations
- Lack of information source
- Lack of domain-specific or private information



Where does the WeAreDeveloper 2023 conference take place?



I'm sorry, but as an AI model developed by OpenAI with a knowledge cutoff in September 2021, I cannot provide real-time information or updates about events happening after that period, including the WeAreDevelopers 2023 conference. To find the location of the conference, I recommend checking the official website or contacting the organizers directly.

Lawyer cites fake cases generated by ChatGPT in legal brief

The high-profile incident in a federal case highlights the need for lawyers to verify the legal insights generated by AI-powered tools.

Published May 30, 2023

Supervised fine-tuning an LLM

- Generating training data is hard and expensive
- Doesn't solve knowledge cutoff issue, only pushed it to a later date
- No data sources
- Hard to debug, is the information coming from foundation training set or from fine-tuning set or is it hallucinated?

From Google documentations

Scenarios to use model tuning

Tuning is required when you want the model to learn something niche or specific that deviates from general language patterns. For example, you can use model tuning to teach the model the following:

- Specific structures or formats for generating output.
- Specific behaviors such as when to provide a terse or verbose output.
- Specific customized outputs for specific types of inputs.

★ **Important:** Currently, model tuning may not improve the model's general knowledge. When tuning a model on a task, if you asked the tuned model a question without including a context, the tuned model may not remember the contexts it was tuned on. We recommend including a context for relevant tasks.



Fine-tuning LLMs is mostly about stylistic changes, not about changing the fundamental capabilities of the model. When fine-tuning, we make the model learn to properly reply to prompts, when to stop generating new tokens, and the style of replies.

Here is a small example of this, comparing the output of falcon-7b foundation model and a fine-tuned version. One can see that the foundation model has the fundamental capabilities, but it lacks a good understanding of the verbosity and structure needed, and also does not know when to stop, and will just continue onwards generating one more token at a time. The fine-tuned model gives exactly what we would expect, a well-structured list of three items, with a short description.

- Providing relevant context in prompt
- Avoid relying on facts provided by LLMs
- Increased transparency due to provided sources
- The answer is as good as the provided context

Retrieval-augmented generation (in-context passing)



Given the provided context:

Nikola Jokic won the NBA finals MVP in 2023. Source: official NBA news

Answer the following question and provide relevant sources.

Do not add any information, just do your job

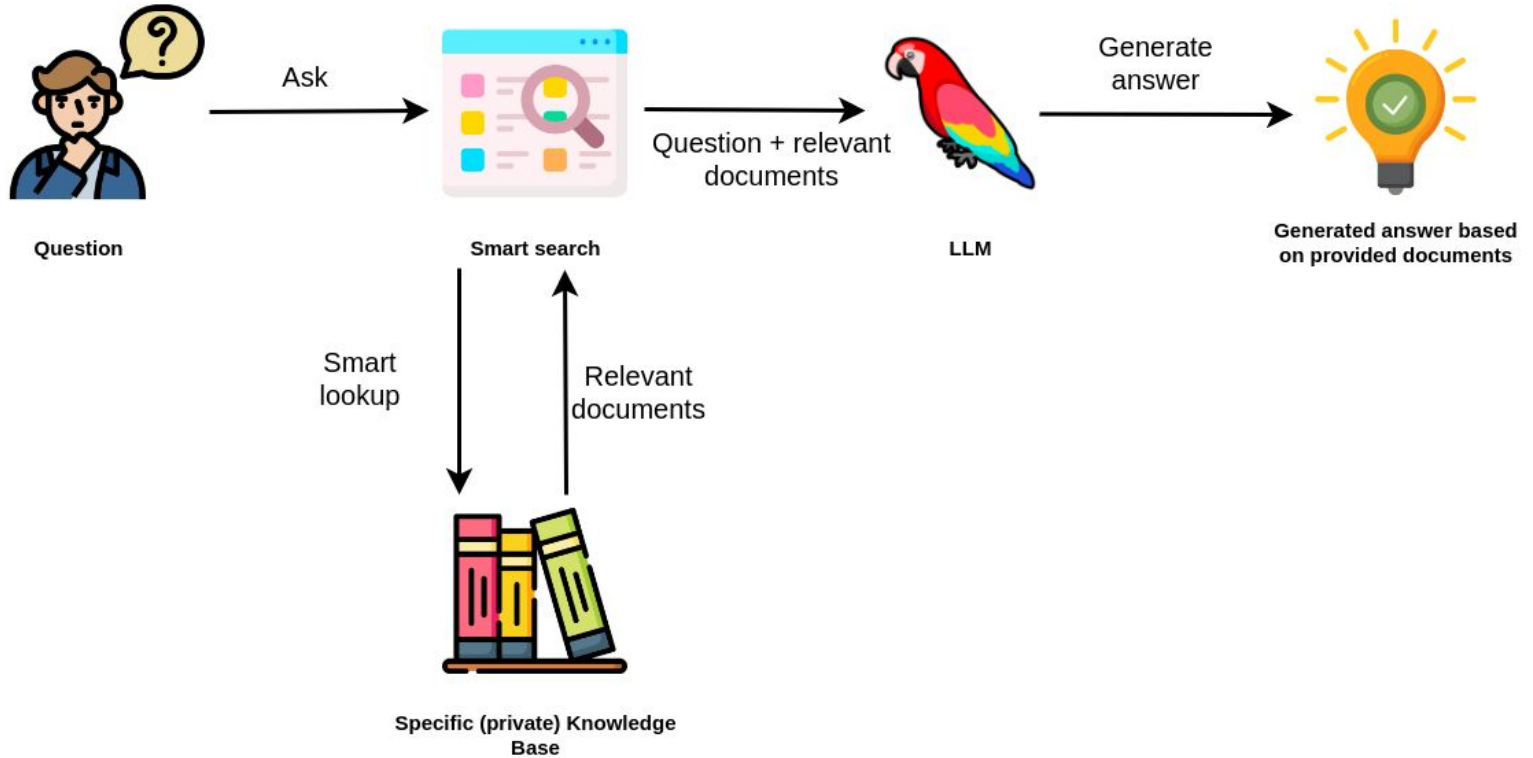
Question:

Who won the NBA finals MVP in 2023?

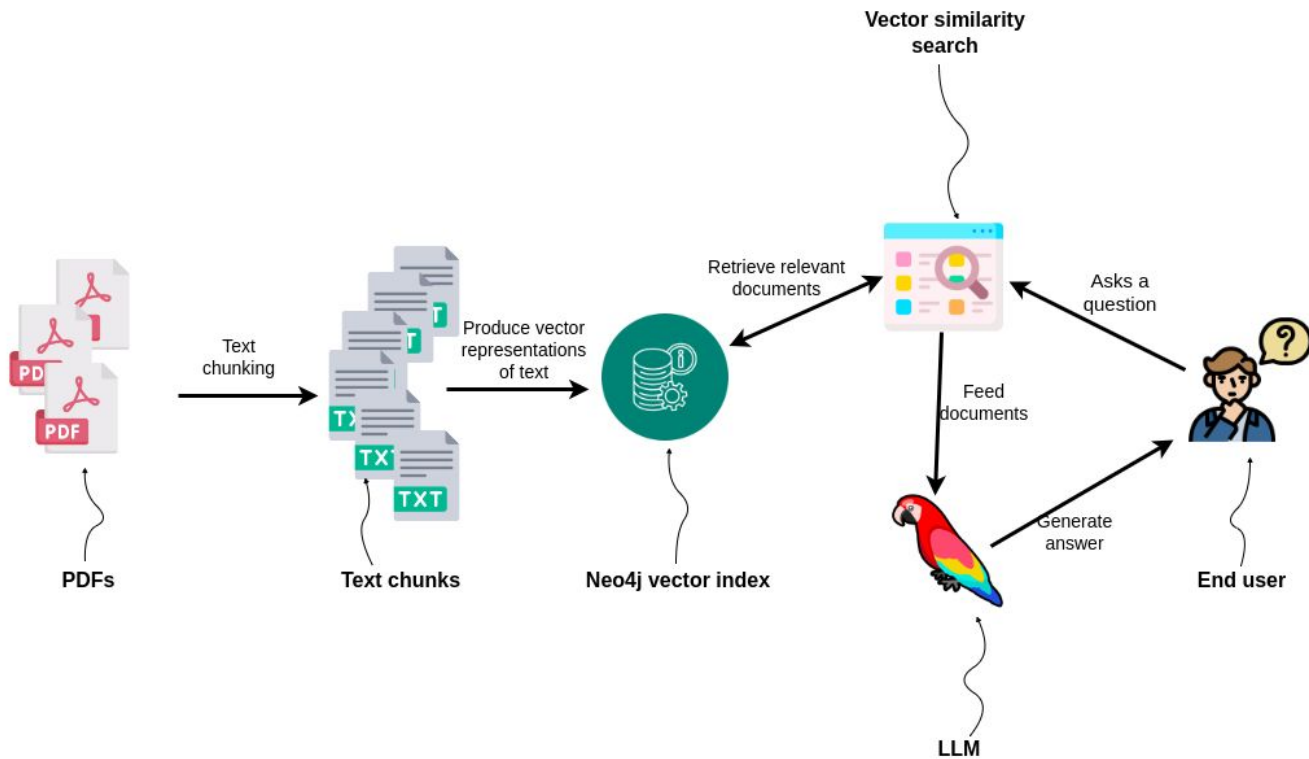


Nikola Jokic won the NBA finals MVP in 2023. The source of this information is official NBA news.

Retrieval-augmented generation (in-context passing) flow



Chat with your PDF application



Vector similarity search

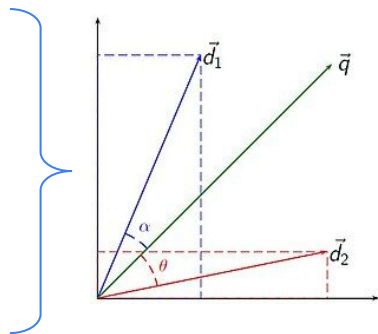
Given a question, find the most relevant documents based on a similarity metric (such as Cosine Similarity) between vector of the question and vectors of contents.

Moving from **keyword** search to **similarity (semantic)** search.

Q: what is text embedding?



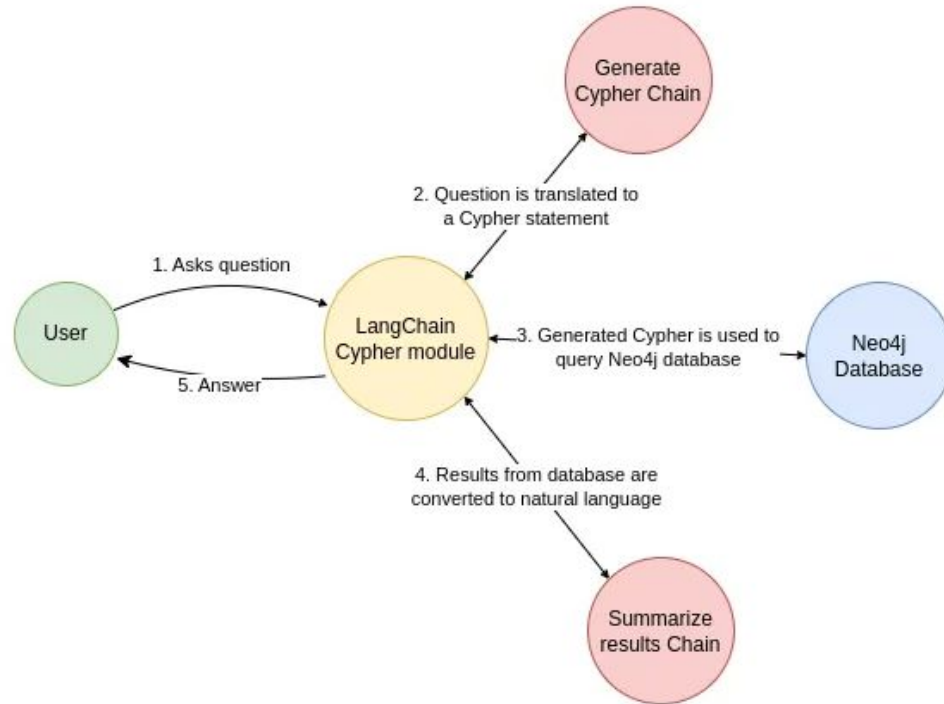
```
embedding [-0.019200351,0.0035306285,0.0
14364655,-0.009949144,0.00115
81815,0.017021084,-0.01998375,
-0.018986698,-0.033643346,-0.0
2841595,0.014685135,0.0141937
32,... Show all]
```



Top K by similarity

abstractId	similarity
456	0.923445
22	0.892114
...	...

Retrieval-augmented generation flow for structured information using generated Cypher statements



Using LLMs to generate queries

- Pass the schema information to LLM along with the question to generate a database query
- Can be enhanced by providing specific query examples in the prompt

This is the schema representation of the Neo4j database.

```
Node properties are the following: [{"properties": [{"property": "name", "type": "STRING"}, {"property": "id", "type": "STRING"}, {"property": "summary", "type": "STRING"}], "labels": "Person"}, {"properties": [{"property": "revenue", "type": "FLOAT"}, {"property": "motto", "type": "STRING"}, {"property": "name", "type": "STRING"}]
```

Relationship properties are the following:

The relationships are the following:

```
[("Person)-[:HAS_PARENT]->(:Person)", "(Person)-[:HAS_CHILD]->(:Person)", "(Organization)-[:HAS_CEO]->(:Person)", "(Organization)-[:HAS_INVESTOR]->(:Organization)", "(Organization)-[:IN_CITY]->(:City)"]
```

Use this schema information to generate a Cypher statement that answers the following question:

Who is the CEO of Neo4j?

Note: Do not answer anything besides the generated Cypher

Sure, the Cypher query to find the CEO of the company named "Neo4j" in your Neo4j database would be:

cypher

Copy code

```
MATCH (o:Organization {name: "Neo4j"})-[:HAS_CEO]->(p:Person)
RETURN p.name
```

Answer generating chain



Giving an LLM presentation with a pop quiz at the end.



Given the provided context:

Nikola Jokic won the NBA finals MVP in 2023. Source: official NBA news

Answer the following question and provide relevant sources.

Do not add any information, just do your job

Question:

Who won the NBA finals MVP in 2023?



Nikola Jokic won the NBA finals MVP in 2023. The source of this information is official NBA news.

Demo time - LLM



```
cypher_chain.run(  
    "How many comics did CAPTAIN AMERICA appear in?")
```

> **Entering new GraphCypherQChain chain...**

Generated Cypher:

```
MATCH (:Character {id: 'CAPTAIN AMERICA'})-[:APPEARED_IN]->(comic:Comic)  
RETURN COUNT(comic)
```

Full Context:

```
[{'COUNT(comic)': 1334}]
```

> **Finished chain.**

```
'CAPTAIN AMERICA appeared in 1334 comics.'
```


About me

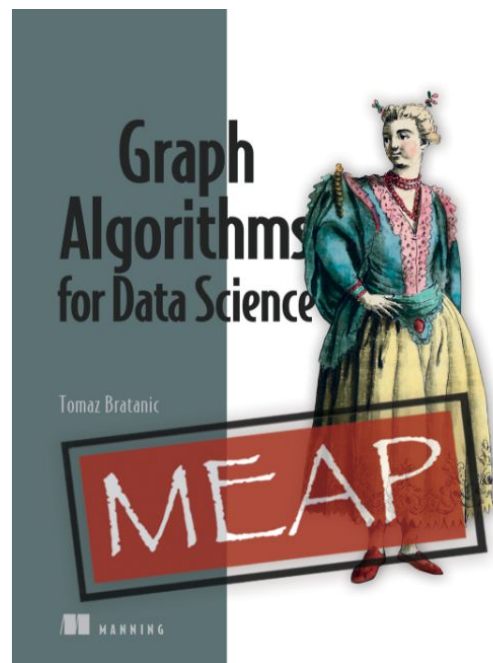
Tomaž Bratanič - Graph & LLM enthusiast

- <https://bratanic-tomaz.medium.com/>
- https://twitter.com/tb_tomaz
- <https://github.com/tomasonjo>
- bratanic.tomaz@gmail.com

Special discount code for 40% off: **ctwnodes40**

<http://mng.bz/vPnM>

Author of:



26