Anticipating Follow-Up Questions in Exploratory Information Search

Graham Wilcock
CDM Interact and University of Helsinki
Helsinki, Finland
graham.wilcock@cdminteract.com

Abstract

The paper describes methods for anticipating follow-up questions in exploratory information search. There are two main cases: information stored in knowledge graphs, and information in unstructured texts such as Wikipedia. In the first case, follow-up questions are anticipated by extracting subgraphs relevant to user queries, passing the subgraphs to an LLM to generate responses. In the second case, entities and their relationships are extracted from the texts and added to short-term knowledge graphs relevant to initial queries. Follow-up questions are then anticipated by extracting subgraphs relevant to subsequent queries and passing the subgraphs to the LLM, as in the first case. The short-term graphs in dialogue memory are often sufficient to answer follow-up questions. If they are not, the described steps are repeated as required.

1 Introduction

Dialogue systems that support users in exploratory information search typically need to handle many follow-up questions. The paper describes methods for anticipating follow-up questions in dialogues for exploratory information search. There are two cases: exploring information stored in knowledge graphs, and exploring information in unstructured texts such as Wikipedia.

The dialogues are exploratory because the users do not yet know where the information is located, or even if it exists. They may not know the structure of the knowledge graphs, or what taxonomy has been used to classify the information into different categories. As a result, users need to keep asking questions as they learn to navigate around different information spaces.

The proposed approach aims to anticipate likely follow-up questions by constructing subgraphs of entities and relationships relevant to current and recent user queries. This can be done while the user is thinking what question to ask next.

If a user is searching existing knowledge graphs, likely follow-up questions can be anticipated by extracting subgraphs relevant to the current user query. The subgraphs are included in prompts to LLMs to generate responses to the user.

If a user is searching unstructured texts such as Wikipedia, there is no knowledge graph from which subgraphs can be extracted. In this case an LLM is prompted to extract entities from the user query, and to extract relevant entities and relationships from the texts, and finally to construct a small short-term knowledge graph from them.

The paper is structured as follows. Section 2 discusses related work. Section 3 summarizes existing methods for generating natural language responses from Wikipedia texts and from knowledge graphs. Section 4 describes new methods for generating subgraphs from existing knowledge graphs and for generating new knowledge graphs from texts. In Section 5 the new methods are used to anticipate follow-up questions in a hybrid retrieval approach combining structured and unstructured retrieval.

2 Related Work

Hogan et al. (2022) is a comprehensive guide to knowledge graphs. Schneider et al. (2022) survey the increasing use of knowledge graphs in NLP.

Sarkar et al. (2020) study methods for extracting subgraphs from DBpedia for use in conversational recommender systems. This is similar to subgraph extraction from knowledge graphs stored in Neo4j graph databases, described in Section 4.1.

A system combining conversational agents with knowledge graphs in Neo4j databases is described by Wilcock and Jokinen (2022). A similar system from Schneider et al. (2023b) aims for synergy between knowledge graphs and conversational agents by bridging the gap between structured and unstructured information retrieval, a topic also addressed here in Section 5 on hybrid retrieval.

Proceedings of the 25th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 103–109
September 18–20, 2024.

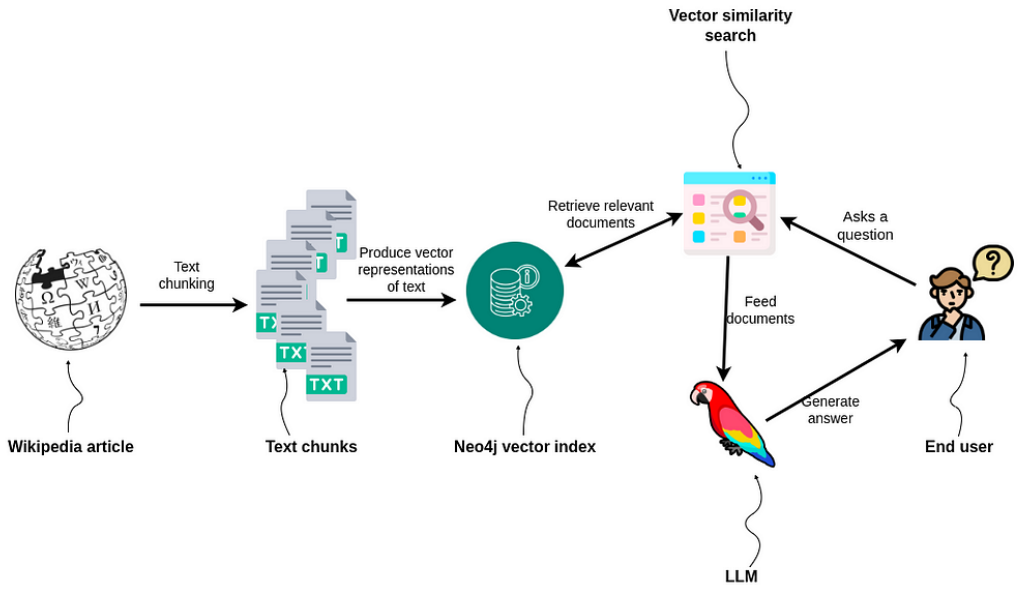

Figure 1: Simple RAG from Wikipedia texts. Image by Tomaz Bratanic, from (Bratanic, 2023a).

Concerning methods for anticipating follow-up questions in exploratory search, Schneider et al. (2023b) mention WikiTalk (Wilcock, 2012), an early robot dialogue system for exploratory search in Wikipedia. Using no knowledge graphs, WikiTalk extracted sets of hyperlinks from Wikipedia articles to transition smoothly between topics by anticipating what the user will ask about next.

Jokinen and Wilcock (2016) proposed a method for anticipation of follow-up topics in Wikipedia search based on hyperlinks and keywords extracted from the current article. This enables anticipating follow-up topics that have no explicit link, and also works for documents without hyperlinks.

The WikiTalk approach of extracting small sets of Wikipedia hyperlinks from the current topic to related topics was motivated by the need at that time to restrict speech recognition vocabulary to a finite list of predicted phrases (Wilcock, 2012). However, the basic idea is similar to retrieving a subgraph or neighborhood of relevant nodes from a knowledge graph, as described in Section 4.1.

RAG (Lewis et al., 2020) is often described as a way of *grounding* LLM responses in the retrieved information, but *conversational* grounding has a long history in dialogue systems research (Traum, 1995; Jokinen, 1996). Grounding is especially important in open-ended conversational exploratory search for navigation in unknown information landscapes (Schneider et al., 2023a).

Theory of Mind errors often arise from failure to build shared knowledge during the dialogue (Wilcock and Jokinen, 2023). Jokinen et al. (2024) investigate the capacity of LLMs to build shared knowledge by classifying grounding-related dialogue acts and by extracting mutually grounded information.

3 LLMs that Generate Responses

RAG enables LLMs to generate natural language responses from retrieved information that is not in their training corpora. This section compares existing methods for RAG from Wikipedia texts and RAG from knowledge graphs.

3.1 Simple RAG from Wikipedia texts

Figure 1 shows a simple RAG application described by Bratanic (2023a) that answers questions based on information from Wikipedia. For a given topic, Wikipedia articles are downloaded and split into texts chunks using LangChain. Vector embeddings of the chunks are generated and stored in a Neo4j database with the texts.

When users ask questions, embeddings of the questions are generated and the most relevant chunks are found by semantic similarity using a Neo4j vector index. The questions and the most relevant chunks are passed to an LLM to generate the answers. Follow-up questions are enabled by using LangChain memory components.

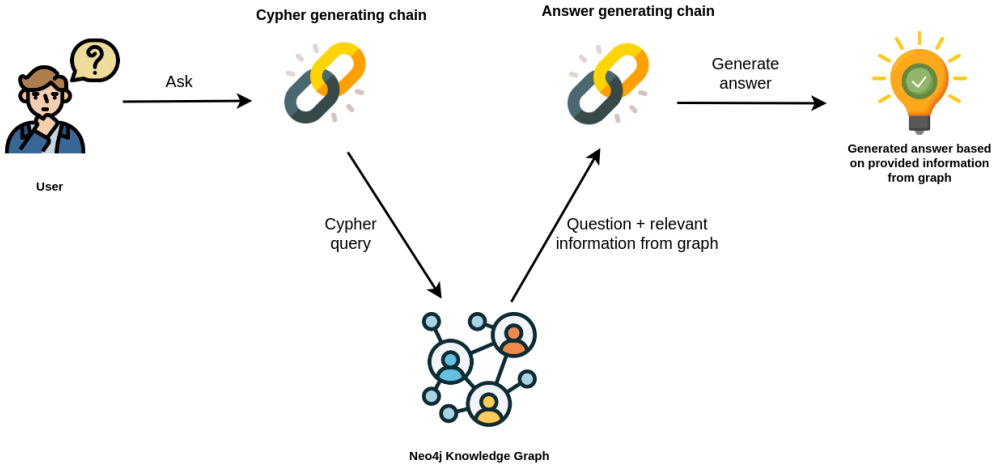

Figure 2: RAG from Knowledge Graphs. Image by Tomaz Bratanic, from (Bratanic, 2023b).

3.2 When simple RAG goes wrong

In order to reduce hallucinations, LLMs can be prompted to avoid making up false facts by using only the information given in the context. However, this can sometimes cause LLMs to avoid telling true facts, by answering as if the facts given in the context are the only true facts in the world.

An example is given by Wilcock (2024), from a *Chat with Wikipedia* application that was given the topic *William Shakespeare*. When asked the question *Did he have any children, grandchildren or other descendants?* the set of most relevant chunks retrieved by RAG did not mention his child Susanna. This caused a conflict between the LLM's own knowledge of Shakespeare and the instructions to use only the information given in the context.

To resolve this conflict, the LLM gave a correct but misleading reply *Yes, William Shakespeare had at least two known children*. The absence of his child Susanna from the context caused the LLM to invent a false justification *There is no direct evidence that he had any other children*.

The follow-up question *Who was Susanna Shakespeare?* caused a new set of chunks to be retrieved and the LLM replied *Susanna Shakespeare was the daughter of William Shakespeare and his wife Anne Hathaway*. It then contradicted its previous reply by adding *Susanna is one of three children known to have been born to Shakespeare and his wife*.

3.3 RAG from knowledge graphs

Recently Neo4j graph databases have been widely used to manage knowledge graphs (Barrasa and Webber, 2023). RAG applications can retrieve information from Neo4j knowledge graphs using Cypher database queries.

Figure 2 from (Bratanic, 2023b) shows RAG from knowledge graphs using two LLMs. The first LLM generates database query code based on the user question. The query retrieves relevant information from the knowledge graph. The second LLM uses the question and the retrieved information to generate the response to the user.

An advantage of RAG from knowledge graphs is that semantic metadata such as taxonomies can be added to the graphs and used to generate more intelligent responses. An example of using knowledge graph metadata in a dialogue system is given by Wilcock (2024). When a user asks for restaurants that serve European cuisine, the graph query finds restaurants serving Italian cuisine. As a taxonomy of cuisines from WikiData was added to the graph, the RAG retrieves the Italian restaurants because Italian cuisine is a subclass of European cuisine in the taxonomy. The LLM gives an intelligent response, explaining that the restaurants serve Italian cuisine which is a type of European cuisine.

4 LLMs that Generate Graphs

We now describe methods for generating subgraphs from existing knowledge graphs and for generating new knowledge graphs from texts.

4.1 Generating subgraphs from graphs

A graph retriever function (Bratanic, 2024) that extracts subgraphs from knowledge graphs in Neo4j graph databases is shown in Figure 3.

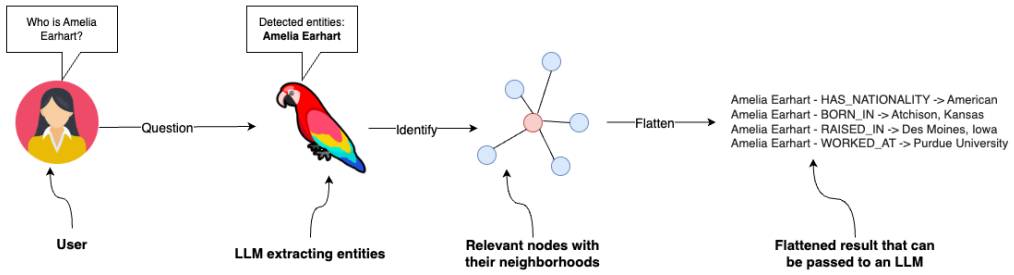

Figure 3: A graph retriever function. Image by Tomaz Bratanic, from (Bratanic, 2024)

The function first extracts entities from the user query. Next, it iterates over the detected entities and uses a Cypher template to retrieve a neighborhood of relevant nodes. The subgraph of relationships between these nodes is converted to a flattened text format that can be passed to an LLM to generate a natural language response to the user.

4.2 Generating knowledge graphs from texts

LLMs can help with knowledge graph construction by analyzing unstructured texts and generating new structured data from them. LLMs must identify the entities mentioned in the texts and identify the relationships between them. They must generate code to create entities and relationships as nodes and relationships in the knowledge graph.

`LLMGraphTransformer` (Bratanic, 2024) helps to construct a knowledge graph by using an LLM to convert texts into graph documents, which can then be imported into Neo4j graph databases. Links to the sources of the texts can be included in the graph documents for provenance checking.

Bratanic (2024) introduces a hybrid approach to retrieval that aims to enhance RAG accuracy by combining vector-based search of unstructured text with structured retrieval of knowledge graph data. The new approach is shown in Figure 4.

To demonstrate the hybrid approach, Bratanic uses `LLMGraphTransformer` to extract entities and relationships from Wikipedia texts about Elizabeth I, convert the texts to graph documents, and import them into a knowledge graph in a Neo4j database.

Elizabeth I - RULED -> England
Elizabeth I - RULED -> Ireland
Elizabeth I - BELONGED_TO -> House Of Tudor
Elizabeth I - PARENT -> Henry Viii
Elizabeth I - PARENT -> Anne Boleyn

Table 1: Generated relationships about Elizabeth I.

Table 1 shows some examples of relationships generated when a user asks about Elizabeth I. The graph retriever has converted the relationships into a flattened text format that is passed in a prompt to an LLM to generate responses to the user.

5 Anticipating Follow-up Questions in Hybrid Retrieval for RAG

The hybrid retrieval approach has been applied to anticipating follow-up questions in exploratory information search from Wikipedia. This short paper does not include evaluation and analysis sections. However, the hybrid retrieval approach can be compared with the simple RAG from Wikipedia texts described in Section 3.1.

In a small trial, `LLMGraphTransformer` was used to construct a knowledge graph about William Shakespeare from Wikipedia texts. Appendix A lists 50 of the generated relationships. Table 2 shows some examples.

William Shakespeare - SPOUSE -> Anne Hathaway
William Shakespeare - PARENT -> Susanna
William Shakespeare - PARENT -> Hamnet
William Shakespeare - PARENT -> Judith
William Shakespeare - PART_OWNER -> Lord Chamberlain's Men

Table 2: Generated relationships about Shakespeare.

When asked the follow-up questions in Table 3, the hybrid retrieval gives the correct information about Shakespeare's children, unlike the error in Section 3.2 with simple RAG from texts.

Question:	When was he born?
Assistant:	*William Shakespeare was born on 23 April 1564.*
Question:	Did he have any children, grandchildren, or descendants?
Assistant:	*Yes, William Shakespeare had three children: Susanna, and twins Hamnet and Judith.*

Table 3: Follow-up questions about Shakespeare.

There are some technical problems in generating knowledge graphs using LLMs. One problem is

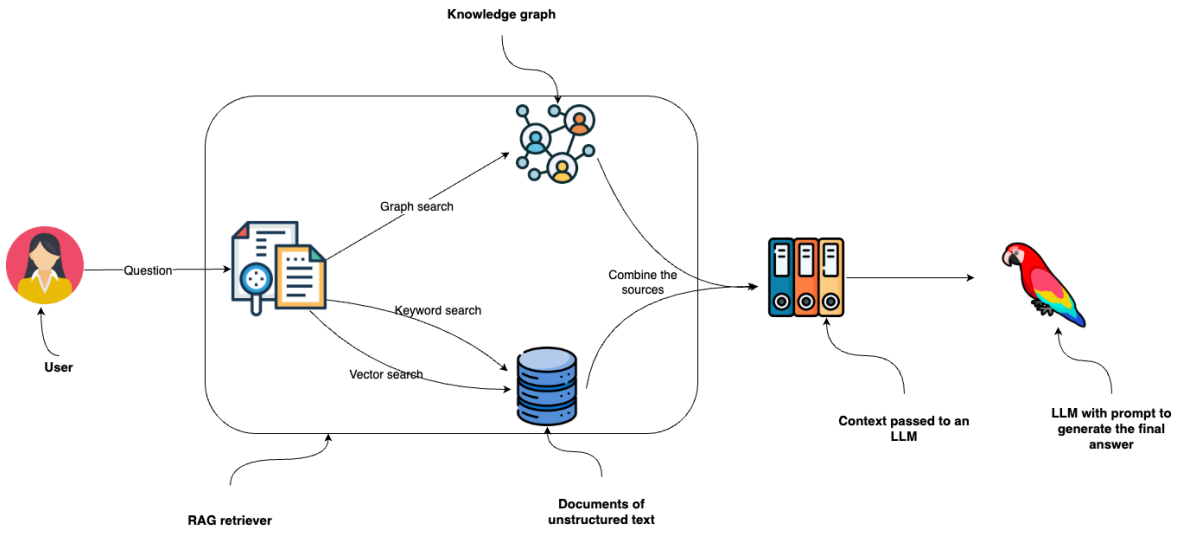

Figure 4: Hybrid Retrieval for RAG. Image by Tomaz Bratanic, from (Bratanic, 2024).

getting the direction of relationships correct. For example in Table 1, PARENT relationships go from Elizabeth I to her parents Henry VIII and Anne Boleyn, but in Table 2, PARENT relationships go from Shakespeare to his children Susanna, Hamnet and Judith. Work to resolve this problem is ongoing.

6 Conclusion

After summarizing existing methods for generating natural language responses from Wikipedia texts and from knowledge graphs, the paper described new methods for anticipating follow-up questions in dialogues for exploratory information search, considering two cases. When exploring information already stored in knowledge graphs, follow-up questions are anticipated by extracting subgraphs that are likely to be relevant to subsequent user queries, and passing the subgraphs to an LLM to generate responses.

When exploring information in unstructured texts such as Wikipedia, entities and relationships are extracted from the texts and used to construct new short-term knowledge graphs relevant to initial user queries. Follow-up questions are anticipated by extracting subgraphs likely to be relevant to subsequent user queries, and continuing as in the first case.

Although there are some problems to be solved in automatic construction of knowledge graphs by LLMs, this kind of approach is attractive. Ongoing work will aim to explore its potential benefits both for anticipating follow-up questions in exploratory information search, and more widely in other areas of spoken dialogue systems.

Acknowledgements

The author thanks Kristiina Jokinen for valuable discussions and fruitful collaboration on dialogue-related topics.

References

Jesús Barrasa and Jim Webber. 2023. *Building Knowledge Graphs: A Practitioner's Guide*. O'Reilly Media.

Tomaz Bratanic. 2023a. LangChain library adds full support for Neo4j vector index. https://neo4j.com/developer-blog/langchain-library-full-support-neo4j-vector-index/.

Tomaz Bratanic. 2023b. neo4j_cypher. https://github.com/langchain-ai/langchain/tree/master/templates/neo4j-cypher.

Tomaz Bratanic. 2024. Enhancing RAG-based applications accuracy by constructing and leveraging knowledge graphs. https://github.com/tomasonjo/blogs/blob/master/llm/enhancing_rag_with_graph.ipynb.

Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutiérrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan F. Sequeda, Steffen Staab, and Antoine Zimmermann. 2022. *Knowledge Graphs*. Morgan & Claypool.

Kristiina Jokinen. 1996. Cooperative Response Planning in CDM: Reasoning about Communicative Strategies. In Anton Nijholt, editor, *Twente Workshop Series in Language Technology*.

Kristiina Jokinen, Phillip Schneider, and Taiga Mori. 2024. Towards harnessing large language models for comprehension of conversational grounding. In *14th International Workshop on Spoken Dialogue Systems Technology (IWSDS 2024)*, Sapporo, Japan.

Kristiina Jokinen and Graham Wilcock. 2016. Double topic shifts in open domain conversations: Natural language interface for a Wikipedia-based robot application. In *Proceedings of the Open Knowledge Base and Question Answering Workshop (OKBQA 2016)*, pages 59–66, Osaka, Japan. The COLING 2016 Organizing Committee.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, pages 9459–9474, Vancouver, Canada.

Rajdeep Sarkar, Koustava Goswami, Mihael Arcan, and John P. McCrae. 2020. Suggest me a movie for tonight: Leveraging knowledge graphs for conversational recommendation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4179–4189, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Phillip Schneider, Anum Afzal, Juraj Vladika, Daniel Braun, and Florian Matthes. 2023a. Investigating conversational search behavior for domain exploration. In *Advances in Information Retrieval: 45th European Conference on Information Retrieval, ECIR 2023, Dublin, Ireland, April 2–6, 2023, Proceedings, Part II*, page 608–616, Berlin, Heidelberg. Springer-Verlag.

Phillip Schneider, Nils Rehtanz, Kristiina Jokinen, and Florian Matthes. 2023b. From data to dialogue: Leveraging the structure of knowledge graphs for conversational exploratory search. In *Proceedings of the 37th Pacific Asia Conference on Language, Information and Computation*, pages 609–619, Hong Kong, China. Association for Computational Linguistics.

Phillip Schneider, Tim Schopf, Juraj Vladika, Mikhail Galkin, Elena Simperl, and Florian Matthes. 2022. A decade of knowledge graphs in natural language processing: A survey. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 601–614, Online only. Association for Computational Linguistics.

David R. Traum. 1995. *A computational theory of grounding in natural language conversation*. Ph.D. thesis, University of Rochester, USA.

Graham Wilcock. 2012. WikiTalk: A spoken Wikipedia-based open-domain knowledge access system. In *Proceedings of the COLING 2012 Workshop on Question Answering for Complex Domains*, pages 57–69, Mumbai, India.

Graham Wilcock. 2024. New technologies for spoken dialogue systems: LLMs, RAG and the GenAI Stack. In *14th International Workshop on Spoken Dialogue Systems Technology (IWSDS 2024)*, Sapporo, Japan.

Graham Wilcock and Kristiina Jokinen. 2022. Conversational AI and knowledge graphs for social robot interaction. In *ACM/IEEE International Conference on Human-Robot Interaction (HRI 2022)*, pages 1090–1094, Sapporo, Japan. Association for Computing Machinery.

Graham Wilcock and Kristiina Jokinen. 2023. To Err Is Robotic; to Earn Trust, Divine: Comparing ChatGPT and Knowledge Graphs for HRI. In *32nd IEEE International Conference on Robot and Human Interactive Communication (RO-MAN 2023)*, pages 1396–1401, Busan, Korea.

A Appendix A

Relationships relevant to William Shakespeare extracted by `LLMGraphTransformer` from Wikipedia texts and imported into a Neo4j knowledge graph. They are shown in a flattened text format that can be passed in prompts to LLMs.
Only 50 relationships are listed here.

```
William Shakespeare - SPOUSE -> Anne Hathaway
William Shakespeare - PARENT -> Susanna
William Shakespeare - PARENT -> Hamnet
William Shakespeare - PARENT -> Judith
William Shakespeare - PART_OWNER -> Lord Chamberlain'S Men
Lord Chamberlain'S Men - NAME_CHANGE -> King'S Men
King James Vi Of Scotland - ASCENSION -> King'S Men
William Shakespeare - FRIEND -> John Heminges
William Shakespeare - FRIEND -> Henry Condell
Shakespeare - FAMILY -> John Shakespeare
Shakespeare - FAMILY -> Mary Arden
Shakespeare - MARRIAGE -> Anne Hathaway
Shakespeare - ACQUAINTANCE -> Ben Jonson
Shakespeare - ACQUAINTANCE -> William Oldys
Shakespeare - ACQUAINTANCE -> George Steevens
Shakespeare - AUTHOR -> Plays
William Shakespeare - AUTHOR -> Plays
Plays - CLASSIFICATION -> Tragedy
Plays - CLASSIFICATION -> History
Plays - CLASSIFICATION -> Comedy
Plays - CLASSIFICATION -> Problem Plays
Plays - CLASSIFICATION -> Romances
Shakespeare - ARRIVAL -> London
Shakespeare - INVOLVEMENT -> The Curtain
Tudor Morality Plays - INFLUENCE -> Shakespeare
Classical Aesthetic Theory - INFLUENCE -> Shakespeare
Classical Aesthetic Theory - DERIVED_FROM -> Aristotle
Classical Aesthetic Theory - DERIVED_FROM -> Plautus
Classical Aesthetic Theory - DERIVED_FROM -> Terence
Rose - SIMILARITY -> Globe
Public Theatres - HAS_FEATURE -> Three Stories High
Public Theatres - HAS_FEATURE -> Open Space At The Center
Public Theatres - HAS_FEATURE -> Polygonal In Plan
Public Theatres - HAS_FEATURE -> Inward-Facing Galleries
Public Theatres - HAS_FEATURE -> Stage
Stage - SURROUNDED_BY -> Platform
Platform - SURROUNDS -> Audience
Stage - HAS_FEATURE -> Rear
Rear - HAS_FEATURE -> Entrances And Exits
Entrances And Exits - USED_BY -> Actors
Entrances And Exits - USED_BY -> Musicians
Public Theatres - HAS_FEATURE -> Upper Level
Upper Level - CAN_BE_USED_AS -> Balcony
Public Theatres - MADE_OF -> Timber
Public Theatres - MADE_OF -> Lath And Plaster
Public Theatres - HAS_FEATURE -> Thatched Roofs
Public Theatres - VULNERABLE_TO -> Fire
Public Theatres - REPLACED_BY -> Globe
Globe - REPLACED_WITH -> Tile Roof
Blackfriars Theatre - ASSOCIATED_WITH -> Shakespeare
```