Local Knowledge Graph Construction from Academic Papers Using Pre-Trained Large Language Models (LLM) Through In-Context Learning

Author: Haoting Chen **Supervisors:** Dr. Sergio J. Rodríguez Méndez, Dr. Pouya G. Omran

0. Background

A **Knowledge Graph (KG)** is a data model capable of storing and representing knowledge [1]. Like graphs in mathematics, a common KG consists of nodes and edges, where a node can represent a named entity [1], e.g., "ANU" and "Canberra", and an edge represents a predicate, e.g. "located in". Every two nodes connected by a predicate form a single piece of knowledge, e.g., ("ANU", "located in", "Canberra"). Such a collection of knowledge triples forms a KG.

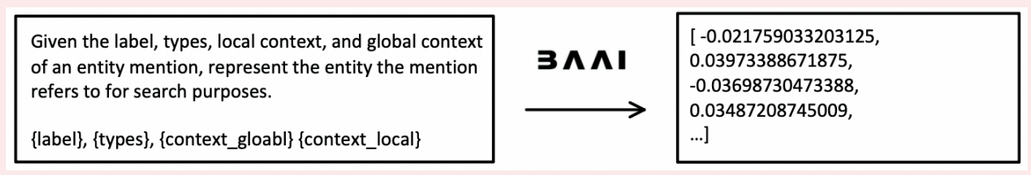
Although KGs can represent knowledge effectively, the **Knowledge Graph Construction (KGC)** task is challenging. It requires significant human effort or machines with a solid natural language understanding. Since the emergence of **Large Language Models (LLMs)** like BERT, researchers have successfully turned them into KGC models by fine-tuning [2]. However, fine-tuning can also consume significant computational power. Instead of changing the models themselves, more advanced LLMs, like GPT-4 and Llama, are already capable of performing a variety of tasks based on the input instructions written in natural language only [3]. Such use of LLMs is called prompt engineering or **in-context learning** [3]. Recent studies of KGC through in-context learning mainly focus on a single or several KGC subtasks [4][5], e.g., named entity recognition (NER). Therefore, our main **objective** is to combine the previous studies and present a functional prompt-based LLM KGC pipeline spanning the entire KGC process. Specifically, our pipeline focuses on taking **semi-structured academic papers** as inputs, while the conversion from raw text or files into semi-structured representation has been widely studied by previous ANU researchers [6]. Our primary **methodology** involves gathering and validating different prompt engineering techniques from the literature for each KGC stage and linking them together, incorporating some new ideas to create a novel KGC pipeline.

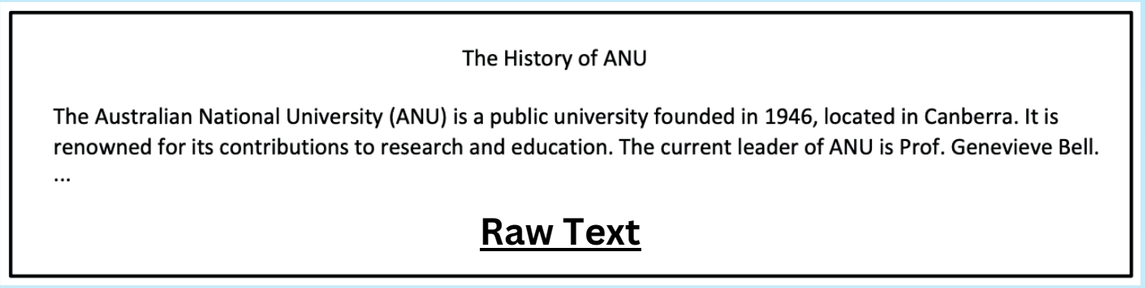
1. Named Entity Recognition & Typing

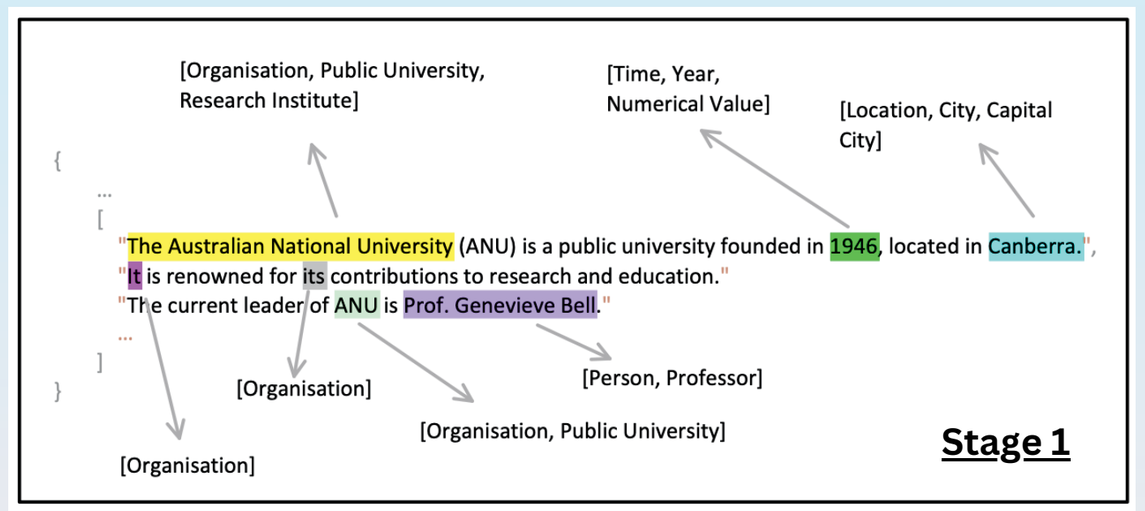
The starting point of KGC is **Named Entity Recognition (NER)** and **Entity Typing (ER)**, where the mentions of any specific objects in the real world [1], like "ANU", "Canberra" are identified and assigned with some types, like "Organisation" and "Location". Traditional NER focuses on extracting named entities that fall into a predefined set of types, and only a single type is assigned to each named entity [1][2]. In contrast, our pipeline allows the LLMs to extract everything they consider a named entity and assign them to one or more potential types. This process is achieved by prompting an LLM decoder like Llama 3 [7] and GPT-4o [8].

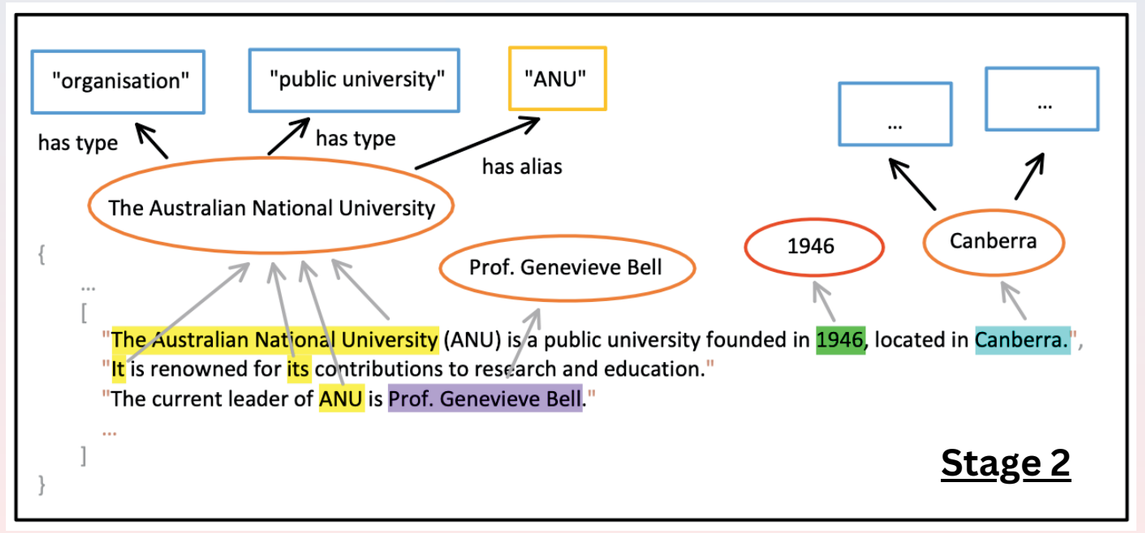
2. Entity Disambiguation & Coreference Resolution

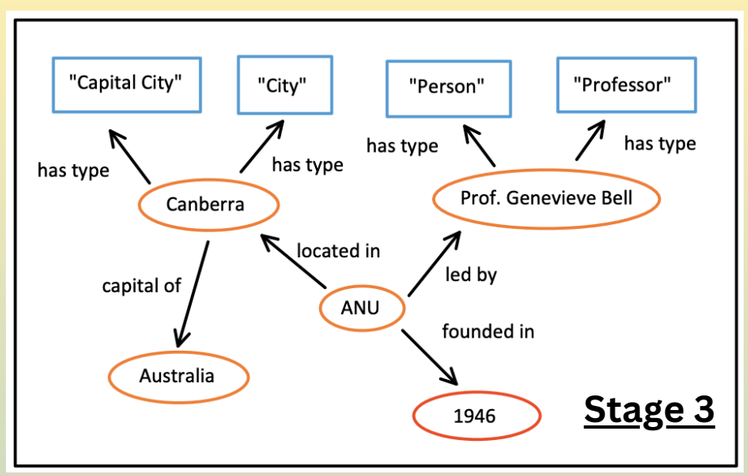
In a text, two different named entity mentions may refer to the same thing, like "ANU" and "The Australian National University", while two identical named entity mentions may refer to different entities, such as two persons named "Jack". **Entity Disambiguation (ED)** and **Coreference Reference (CR)** are the tasks of grouping and disambiguating these mentions [1]. This process is achieved by prompting an LLM encoder, BAAI's bge-base-en-v1.5 [9], to generate a context-sensitive **vector representation**, embedding, for each mention With embeddings, we can easily know if two mentions are similar or different by computing their cosine similarity.

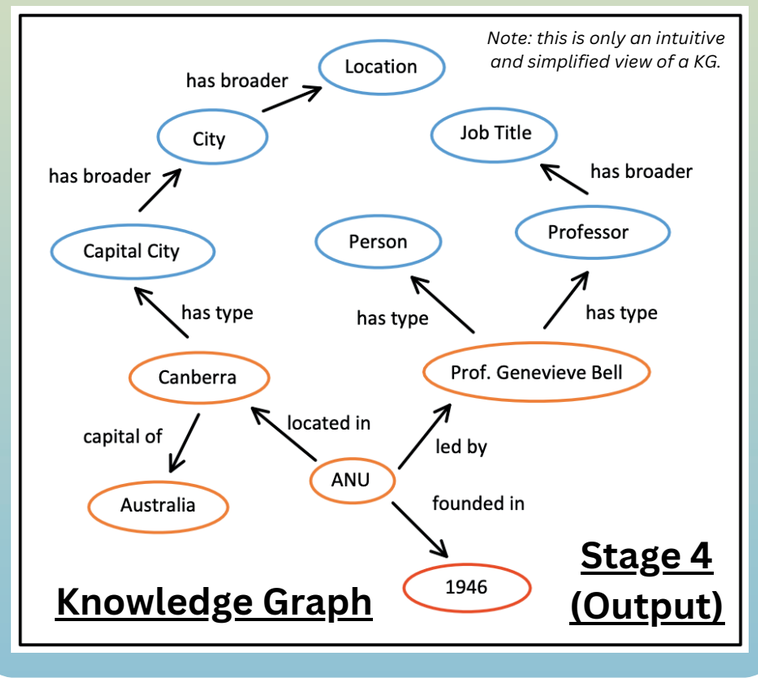










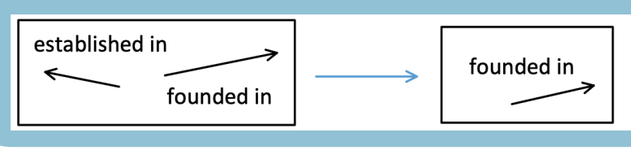


3. Relation Extraction

Relation Extraction (RE) aims to create labelled and directed edges, called predicates, between any two related entities in the knowledge graph based on a given text [1]. Previous studies also mainly focus on connecting entities with a set of predefined predicates [1][2]. Recently, Zhang and Soh's study [5] has introduced the concept of open RE, which gives freedom to the LLMs to extract any relations. To achieve RE, we first prompt the LLMs to extract relations between the entities within a local text chunk (either a single sentence, paragraph, or section). For document-level RE, we first prompt the LLMs to generate a summary of the paper since the original paper may be longer than the LLMs' input limits. Given the summary and for two entities, we then prompt the LLMs to predict their relations.

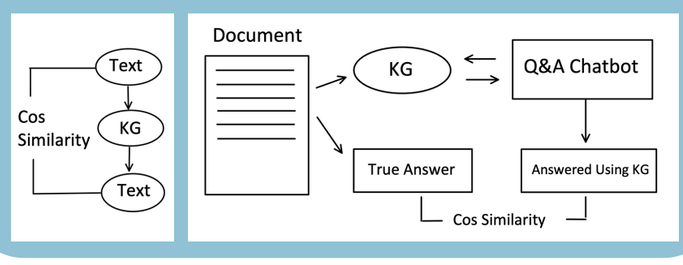
4. Schema Generation

The entity types and predicates are important parts of the **schema** of a KG. Traditionally, this schema is created manually before any KGC tasks [1]. Therefore, the traditional NER and ER may only focus on extracting named entities and relations that satisfy the schema. Since we have gone through an open NER and RE without aligning them with any schema, our current KG schema has yet to be concrete. Our types are currently "flattened" and both our types and predicates may contain redundancy. Like in ED and CR, as shown in 3, these steps aim to group and distinguish types and predicates. In addition, we want our types to form a hierarchy. Therefore, we first prompt the LLMs to generate a potential parent type for each type and embed both the type itself and its potential parent type. If the embedding of the potential parent of Type A is similar to the embedding of Type B, then Type B is likely the parent of Type A.



5. Evaluation

Current public datasets that align input documents with output KG have their own predefined schemas. Since our schema is generated on the fly, it may not be a good idea to use their output KG as our ground truth for comparison and evaluation since our schemas are already different. Even without a direct target output to compare against, we can still evaluate our KGC pipeline by reversing our KGs back to texts to test how much and how accurately the KG can capture the original content. The second method is to use the TriviaQA dataset [10], which includes documents, questions, and ground truth answers, to evaluate how effectively our KG can be used to answer questions.



6. Contribution and Application

The main contribution of this research is to enrich the ANU Academic Scholarly Knowledge Graph (ASKG). Currently, it mainly contains semi-structured representation of academic papers instead of their full knowledge graph representation [11]. For broader applications, as demonstrated in the testing section, the local KG for a paper can be used by a Q&A chatbot to quickly and accurately locate and retrieve relevant facts to answer user questions about a paper. This is possible because the KG stores structured information that is more machine-readable than raw text. The local knowledge graph may also serve as a mind map, virtually supporting researchers who want a quick overview of a specific paper.

The poster design was created using a template provided by Canva (www.canva.com).
[1]Hogan, A., "Knowledge Graphs", arXiv e-prints, Art. no. arXiv:2003.02320, 2020. doi:10.48550/arXiv.2003.02320.
[2]Pan, S., Luo, L., Wang, Y., Chen, C., Wang, J., and Wu, X., "Unifying Large Language Models and Knowledge Graphs: A Roadmap", arXiv e-prints, Art. no. arXiv:2306.08302, 2023. doi:10.48550/arXiv.2306.08302.
[3]Sahoo, P., Singh, A. K., Saha, S., Jain, V., Mondal, S., and Chadha, A., "A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications", arXiv e-prints, Art. no. arXiv:2402.07927, 2024. doi:10.48550/arXiv.2402.07927.
[4]Papaluca, A., Krefl, D., Mendez Rodriguez, S., Lensky, A., and Suominen, H., "Zero- and Few-Shots Knowledge Graph Triplet Extraction with Large Language Models", arXiv e-prints, Art. no. arXiv:2312.01954, 2023. doi:10.48550/arXiv.2312.01954.
[5]Zhang, B. and Soh, H., "Extract, Define, Canonicalize: An LLM-based Framework for Knowledge Graph Construction", arXiv e-prints, Art. no. arXiv:2404.03868, 2024. doi:10.48550/arXiv.2404.03868.
[6]Zhang, B, Rodríguez-Méndez, S. J. and Omran, P. G., "ASKG: An approach to enrich scholarly knowledge graphs through paper decomposition with deep learning," CEUR Workshop Proceedings, vol. 3632, 2023.
[7]Dubey, A., "The Llama 3 Herd of Models", arXiv e-prints, Art. no. arXiv:2407.21783, 2024. doi:10.48550/arXiv.2407.21783.
[8]OpenAI, "GPT-4 Technical Report", arXiv e-prints, Art. no. arXiv:2303.08774, 2023. doi:10.48550/arXiv.2303.08774.
9]Xiao, S., Liu, Z., Zhang, P., Muennighoff, N., Lian, D., and Nie, J.-Y., "C-Pack: Packed Resources For General Chinese Embeddings", arXiv e-prints, Art. no. arXiv:2309.07597, 2023. doi:10.48550/arXiv.2309.07597.
[10]Joshi, M., Choi, E., Weld, D. S., and Zettlemoyer, L., "TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension", arXiv e-prints, Art. no. arXiv:1705.03551, 2017. doi:10.48550/arXiv.1705.03551.
[11]Jia, R., Zhang, B., Rodríguez Méndez, S. J., and Omran, P. G., "Leveraging Large Language Models for Semantic Query Processing in a Scholarly Knowledge Graph", arXiv e-prints, Art. no. arXiv:2405.15374, 2024. doi:10.48550/arXiv.2405.15374.