June 2024

Plan Generation from Unstructured Documents Through Transformer-Based Extraction of Knowledge Graphs

Wolfgang Maass
German Research Center for Artificial Intelligence (DFKI), wolfgang.maass@dfki.de

Cicy K. Agnes
German Research Center for Artificial Intelligence (DFKI), cicy.agnes@dfki.de

Amin Harig
German Research Center for Artificial Intelligence (DFKI), amin.harig@dfki.de

PLAN GENERATION FROM UNSTRUCTURED DOCUMENTS THROUGH TRANSFORMER-BASED EXTRACTION OF KNOWLEDGE GRAPHS

Completed Research Paper

Wolfgang Maass, German Research Center for Artificial Intelligence (DFKI) and Saarland University, Saarbrücken, Germany, wolfgang.maass@dfki.de

Cicy K. Agnes, German Research Center for Artificial Intelligence (DFKI), Saarbrücken, Germany, cicy.agnes@dfki.de

Amin Harig, German Research Center for Artificial Intelligence (DFKI), Saarbrücken, Germany, amin.harig@dfki.de

Abstract

Planning for complex tasks is a key task for knowledge workers that is often time-consuming and depends on the manual extraction of knowledge from documents. In this research, we propose an end-to-end method, called PlanKG, that: (1) extracts knowledge graphs from full-text plan descriptions (FTPD); and (2) generates novel FTPD according to plan requirements and context information provided by users. From the knowledge graphs, activity sequences are obtained and projected into embedding spaces. We show that compressed activity sequences are sufficient for the search and generation of plan descriptions. The PlanKG method uses a pipeline consisting of decoder-only transformer models and encoder-only transformer models. To evaluate the PlanKG method, we conducted an experimental study for movie plot descriptions and compared our method with original FTPDs and FTPD summarizations. The results of this research has significant potential for enhancing efficiency and precision when searching and generating plans.

Keywords: Knowledge Graphs, Modeling, Plan Generation, Text Embeddings, Large Language Models, Transformer models, Knowledge Management.

1 Introduction

For many decades IS researchers studied how to capture, represent, and model different types of knowledge, including the transformation of data to knowledge (Dalkir, 2017), conceptual modeling (Chen, 1976) and its support for knowledge management (Ale et al., 2014; Mineau, Missaoui, and Godinx, 2000) and natural language processing (Montes et al., 2008). Procedural knowledge (know-how) is particularly important because it turns implicit and explicit knowledge into tangible resources that can be shared with others (Alavi and Leidner, 2001). Products are physical manifestations of knowledge, while services are procedural manifestations of knowledge.

Early work on knowledge management represented knowledge by encoding it into text documents, storing it in content management systems, and accessing it using sophisticated search engines (Maier and Hadrich, 2011). Because of the heterogeneity and ambiguity of text, search results are only pointers to documents or fragments of documents. Therefore, the results require substantial cognitive work to make it accessible for planning and other knowledge tasks. For natural language processing tasks, transformer architectures have led to significant improvements (Vaswani et al., 2017) in areas, such as summarization and extraction of knowledge from text. Beyond unstructured text, early research

investigates the use of transformer models for extracting structured knowledge graphs from text documents (e.g., (Khorashadizadeh et al., 2023; Trajanoska, Stojanov, and Trajanov, 2023))). The behavior of transformer models is traditionally controlled by modifying text used as input (prompt engineering). The ambiguity of natural language reduces control, so that more formalized input structures become important. In this paper, we present PlanKG, an approach that extracts knowledge graphs (Singhal et al., 2012) from textual descriptions and subsequent compressions that is used as contextual input for the task of creating plan descriptions.
Our research is driven by the following questions:

- **RQ1**: How to extract knowledge graphs from textual descriptions of plans by large-language models?
- **RQ2**: Is the similarity of textual plan descriptions improved by knowledge graph representations compared to text documents?
- **RQ3**: Are created plan descriptions improved based on knowledge graph representations compared to text documents?

This paper proceeds as follows. Section 2 reviews related work in planning and knowledge graph generation. Section 3 proposes the PlanKG method. The method is applied to movie plots, as an easily understandable example of plans, in Section 4, and the results are discussed in Section 5. Section 6 concludes the paper.

2 Related Work

In this section, we consider existing research on plans, including scripts and movie plots, text summarization, and embeddings.

2.1 Plan generation and extraction

Four generic activities are basic for processing knowledge: creating, storing/retrieving, transferring, and applying (Alavi and Leidner, 2001). The creation of procedural knowledge in the form of plans are used as boundary objects for exchange with other people. A plan is "any hierarchical structure of intended actions, through which an individual anticipates acting to achieve a goal" (Miller, Galanter, and Pribram, 1960). Planning is the activity of creating plans that satisfy intentions and planning constraints (Coyne and Gero, 1985). Plans in the sense of problem-solving are descriptions of procedures intended to transform initial states into solution states (Simon and Newell, 1971). While complex plans contain parallel sub-plans, looping, and conditional branching, one-dimensional sequential plans dominate practical work, even in domains such as construction (Parfitt and Sanvido, 1993) and surgery (Borchard et al., 2012). Plans are sequences of actions or steps necessary to achieve a specific goal or objective. For instance, checklists are a simple, structured form of sequential plans that can be very effective as an informational job aid used to reduce failure by compensating for potential limits of human memory and attention. Many documents of police case files on accident sequences (Clarke, Forsyth, and Wright, 1999) or medical treatment reports (Walker, Reshamwalla, and Wilson, 2012) are descriptions of activity sequences. Scripts as simpler forms of plans are stereotypical sequences of actions as they occur in familiar situations (Schank and Abelson, 1975). A special form of narrative-oriented scripts are movie plots that help to organize the story of a movie into a coherent and engaging sequence of events and activities (McKee, 1997).
To capture procedural knowledge such as plans, information systems have used rule-based approaches and different types of formal logic (KBS, Prolog, SemanticWeb, Linked Data, etc.) (e.g., (Turban and Watkins, 1986)). These approaches have provided actionable results (e.g., (Yoon, Guimaraes, and O'Neal, 1995)). Problem-solving and planning are based on methods and formalisms for representing and using procedural knowledge (LaValle, 2006). Planning is important because, if future actions cannot be predicted, then, the actions that can be taken remain a function of the current state of an information system (LaValle, 2006).

Formal planning has successfully been used in multi-agent systems for proper coordination among agents, such as in robotic football. Although formal planning on a small scale has shown excellent results, the results do not scale well, because formal planning is rarely modeled and used in practice.
Knowledge graphs were introduced by Google as graph structures that create relationships between data elements (Singhal et al., 2012). Wordnet, Freebase, and YAGO are large knowledge graphs (X. Chen, S. Jia, and Xiang, 2020). Knowledge graphs represent knowledge of the real world with nodes that represent entities and edges that represent relationships between them (Hogan et al., 2021). Knowledge graphs have been developed based on research on databases, the semantic web, natural language processing, machine learning, knowledge representation, and other areas (Guti.rrez and Sequeda, 2021). Knowledge graphs attracted interest of both academia and industry for their applications in tasks such as recommendations (Guo et al., 2020), semantic search (K. Wang et al., 2020), question answering (Fu et al., 2020), natural language generation (Koncel-Kedziorski et al., 2022) among others. Since the combination of structured and unstructured knowledge can benefit a wide variety of natural language processing tasks, the use of knowledge graphs is gaining attention in knowledge acquisition and application tasks in natural language processing (Schneider et al., 2022). In this context, embeddings of both knowledge graphs and plain text are relevant to our research since embeddings can be used to capture their underlying semantics and map them into a continuous multidimensional vector space and hence allows for making both these document types comparable on a semantic level (cf. (Z. Wang et al., 2014)).
Humans use mental models to understand and represent the real world in which they live. They are a component of a person's cognitive structure (Forstmann, Wagenmakers, et al., 2015). In making sense of their world, humans use plans for effective interaction with the environments around them. They are fundamental to human thinking and behavior by connecting attitudes and intentions with behavior (Ajzen, 1991). According to the Merriam-Webster dictionary, plan, design, plot, scheme, and project "mean a method devised for making or doing something or achieving an end". In this paper, we use the terms plan and plot interchangeably to describe a sequence of activities that is either performed in the future or has occurred in the past. For instance, the description of a movie plot is used in both ways.
Transformer-based language models show impressive results for automatic text summarization, which is the process of balancing the trade-off between reducing a document while preserving essential information content and meaning (Pilault et al., 2020). However, the generative aspect of LLMs also enables the creation of plan descriptions, such as checklists. Planning is important in Artificial Intelligence because it investigates computational systems for determining a reasonable procedure for progressing from an initial state to a solution state (Fikes and Nilsson, 1971). An important application for which planning has been extensively used is for motion planning in robotics (Garrett et al., 2021). Recently, transformer models have been used to generate action plans for embodied agents, where each action step is semantically translated into an admissible action (M. Chen et al., 2021; Huang et al., 2022). Currently, LLMs show impressive results for simple planning tasks, but poor results for action-diverse natural language texts (Olmo, Sreedharan, and Kambhampati, 2021; Valmeekam et al., 2022; Xie et al., 2023). Given this new territory of LLM for planning tasks, we will focus on sequential planning tasks, with more complex planning tasks as future work.

2.2 Embeddings

Embeddings are dimensionality reduction methods that project input data with different types into a vector space. Different embeddings can be used, such as linear methods (e.g., Principal component analysis (PCA), singular value decomposition (SVD), Linear Discriminant Analysis (LDA)), as well as non-linear methods, such as global structure-based methods (e.g., Multi-Dimensional Scaling (MDS)) or local structure-based methods (e.g., t-Distributed Stochastic Neighbor Embedding (t-SNE), Uniform manifold approximation and projection (UMAP)). (For an overview cf. (W. Jia et al., 2022)).
Embedding techniques, such as Word2Vec, (Mikolov et al., 2013) capture semantic relationships between words. Text preprocessing techniques, including stop word removal and lemmatization, improve the quality of text embeddings. Similarly, the quality of embeddings is improved by using abstracts instead of keywords (Alexandrov, Gelbukh, and Rosso, 2005). Long Short-Term Memory

(LSTM) models (Hochreiter and Schmidhuber, 1997) in combination with attention mechanisms (Vaswani et al., 2017) have been used for sentence embeddings (Lin et al., 2017), followed by Transformer models (Brown et al., 2020; Huang et al., 2022; Vaswani et al., 2017). Transformer models are generalizations of Principal Component Analysis (PCA) that are capable of learning non-linear functions that are built with deep neural networks and self-attention mechanisms and capture the effect of semantic associations with long intervals. Transformer models consist of an encoder and decoder, although recent transformers focus on decoder-only architectures (Chowdhery et al., 2022). The encoder consists of multi-head attention, short-cut connection, layer normalization, and feed-forward. In contrast to pre-trained transformers, such as BERT (Devlin et al., 2018), generative pre-trained transformers (GPT) use unsupervised generative pre-training and supervised adaptation achieving promising results in the zero- and one-shot settings (Brown et al., 2020).

Different embedding methods have been proposed for knowledge graphs, such as TransE (Wang et al., 2014). These approaches focus on embedding entities and encoding relations as distances that are optimized with respect to a given loss function. For embedding unstructured texts, SBERT is a leading sentence embedding model that maintains semantically meaningful embeddings (Reimers and Gurevych, 2019). It uses a siamese model architecture, adds a pooling operation to the output of a BERT model to derive a fixed-sized sentence embedding and supports cosine distances between embeddings of sentences. For interpretation of embeddings, clustering is applied that requires reduction of dimensionality by using non-linear manifold aware dimension reduction by UMAP (McInnes et al., 2018) and subsequent clustering in lower dimensions by, for instance, k-Means (Jain, Murty, and Flynn, 1999).

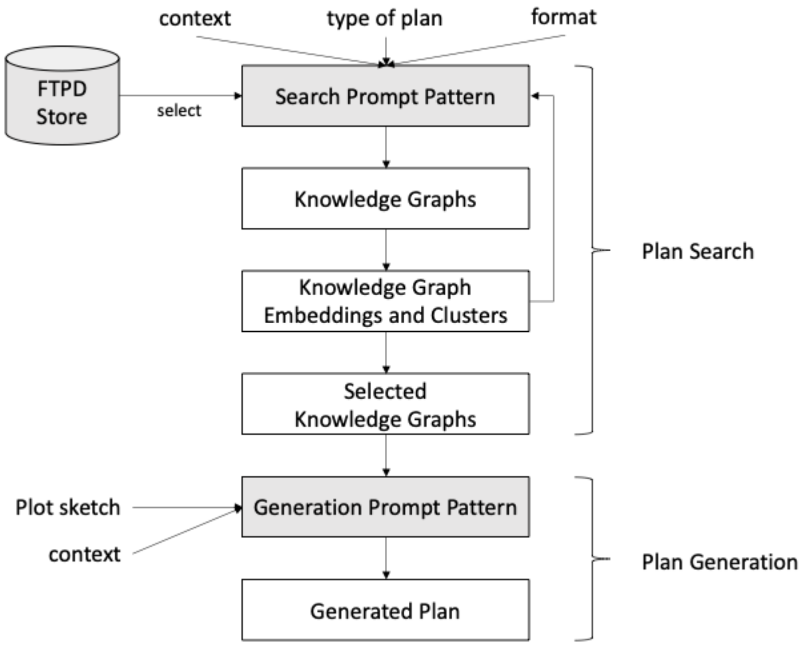

Figure 1. PlanKG method

3 PlanKG Method

This section presents the PlanKG method (cf. Figure 1), which enables controlled search (cf. Figure 2) and generation of sequential plans from unstructured texts by using knowledge graphs extracted by autoregressive language models. Given a set of full-text plan descriptions (FTPD Store), the PlanKG method (1) starts with the application of a search prompt pattern to create a specified type of knowledge graph in a specified format (cf. 3.1). Resulting knowledge graphs are embedded by using autoregressive models. Autoregressive models predict the conditional probability of the next token given the previous tokens. The output tokens are generated sequentially (Paa. and Giesselbach, 2023). Autoregressive language models generate the next words based on the previous words by learning statistical patterns and dependencies. They can handle variable context length, have contextual understanding, and can capture sequential dependencies maintaining the logical flow of text. The plan generation phase uses knowledge graph representations as input with user requirements (plot sketch) and context information (optional). The generated plan is the output of PlanKG (cf. Figure 1).

3.1 Plan extraction

The PlanKG method takes full-text plan descriptions (FTPD) as input and extracts knowledge graphs by leveraging relational machine learning (Nickel et al., 2015) with few-shot autoregressive language models (e.g., (Brown et al., 2020; Chowdhery et al., 2022). These large language models (LLM) provide a means for extracting knowledge graph representations. Prompts are designed for different knowledge graph formats, although expressive representation languages, such as STRIPS, are beyond the scope of current large language models (LLM). The output depends on proper prompt engineering (Liu and Chilton, 2022) by using a pattern as follows:

<Format **pm** and **cm** in **fm** from:", **FTPD** >

Various knowledge graph representation formats (format marker fm) can be extracted by using LLMs, e.g. N-Triples, Turtle, JSON-LD, and RDF/XML. In this reserach, we are using RDF/XML due to its expressiveness and wide use in research (Manola, E. Miller, McBride, et al., 2004). Context markers cm are, for instance, genre, disease, or sports. Sequential and complex planning markers pm are distinguished and we will focus on sequential planning markers in the following (cf. Table 1: pm"named events"). Table 1 gives an example of a knowledge graph extracted from an FTPD (movie plot Titanic) formatted as RDF/N3.

The use of context information depends on the domain and is part of prompt engineering (Liu and Chilton, 2022). The use of PlanKG is iterated until an optimal knowledge graph embedding is reached. During iterations, prompts are adjusted if clusters do not separate according to given context markers. For instance, a search prompt often results in a knowledge graph with insufficient details. Adding the request for additional context information enriches the knowledge graph and often improves the cluster quality. This iterative process requires collaboration with domain experts.

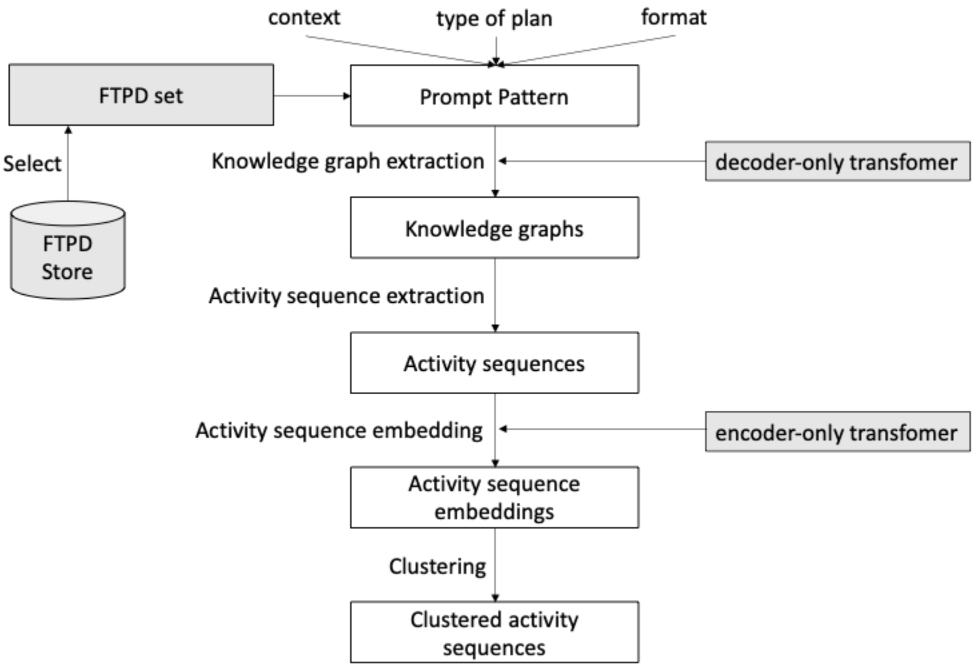

Figure 2. PlanKG: plan search phase

Text Document Title: Titanic
FTPD: In 1996, treasure hunter Brock Lovett and a team from the Russian research ship Keldysh are in the North Atlantic in search of the missing diamond necklace "Heart of the Ocean", which is said to be on board the legendary Titanic. The team manages to salvage a safe from a millionaire's suite. Brock sees fame and fortune in his mind's eye, but instead of the diamond, inside is just a nude drawing dated April 14, 1912of a young woman wearing said piece of jewellery. The 100-year-old Rose Calvert found out about the find in a television report. She immediately contacts the treasure hunters and claims to be the woman depicted. Brock's confidant Lewis Bodine doesn't trust her because she was an actress in the 1920s under the name "Rose Dawson". His suspicions are reinforced by the fact that Rose DeWitt Bukater, who Rose Calvert claims ...
Prompt Create a RDF in N3 representation with genre description and events for: FTPD
Planning marker: events
Format marker: RDF\$N3 format
Context marker: genre
ex:Event1 rdf:type ex:TreasureHunting ; ex:tookPlaceIn ex:NorthAtlantic ; ex:hasYear "1996" . ex:Event2 rdf:type ex:Salvage ; ex:involvedTeam ex:Team1 ; ex:hasYear "1996" . ex:Event3 rdf:type ex:Discovery ; ex:discoveredBy ex:RoseCalvert ;

ex:hasYear "1996" . ex:Event4 rdf:type ex:Claim ; ex:claimedBy ex:RoseCalvert ; ex:hasYear "1996" . ex:Event5 rdf:type ex:Contact ; ex:contactedBy ex:RoseCalvert ; ex:hasYear "1996" . ... ex:Genre1 rdf:type ex:Romance . ex:Genre2 rdf:type ex:Drama . ex:Genre3 rdf:type ex:Historical . ...
Activity sequence: TreasureHunting, Salvage, Discovery, Claim, Contact, Invitation, Introduction, Argument, Escape, Rescue, Reunion, Flashback, Throw
Genres: Romance, Drama, Historical

Table 1. Prompt markers extracted knowledge graph in RDF$\backslash$N3, and extracted activity sequence.

3.2 Activity sequence extraction

For detailed FTPDs, knowledge graphs often become large due to information that is not directly related to activities. This leads to weak embeddings and blurred clusters and an overall search method that lacks control. For better control of the search and generation processes in PlanKG, complete knowledge graphs are reduced to simpler *activity sequences* containing the time-ordered list of event names (cf. example in Figure 2). Due to the emphasis on activities, PlanKG extracts activities and keeps the temporal order by *activity sequences* (cf. Table 1). An activity sequence is a highly compressed knowledge graph representation of an FTPD. In an event triple of a RDF representation of a knowledge graph, a subject refers to an activity. The subject can be an agent who performs an activity of an event in which an activity is performed while the predicate is the activity descriptor. Both are connected by the generic RDF relation *rdf:type*.

3.3 Embeddings and clustering of activity plans

A large number of text embedding methods exist with dimensionalities ranging from 300 to 12288.[1] With increasing dimensions the number of data points required for good results grows exponentially (LeCun, Bengio, and Hinton, 2015). Therefore, embedding models need to be adjusted to the complexity of knowledge graphs. In the current implementation of PlanKG, SBERT is used with a dimensionality of 768 (Reimers and Gurevych, 2019). Dimensionality reduction is achieved by applying non-linear manifold-aware dimension reduction (e.g., UMAP) followed by clustering (e.g., k-means or DBScan).

3.4 Generation of plan descriptions

Plans are related to a domain in which they become meaningful. For the generation of a plan, a context needs to be defined. Frameworks, such as COBIT or ITIL, are practical plan descriptions consolidated from many individual plans for the governance of information systems and service systems. The same

[1] https://huggingface.co/spaces/mteb/leaderboard

holds for checklists in aviation and medicine. In PlanKG, the set of plan descriptions provides the knowledge context from which a plan description is derived. The generation of plan descriptions is performed in two steps:

1. Build plan context
2. Generate plan

In PlanKG, a plan context is built in two steps: (1) selection of plan descriptions and (2) LLM-based extraction of knowledge graphs. The first step can be conducted by traditional database queries or by text queries. After selecting a subset of plan descriptions, knowledge graphs are extracted by application of the instantiated search prompt pattern (cf. section 3.1). The clustering of embedding of activity sequences extracted from knowledge graphs supports the identification of plan descriptions that are similar to a given search prompt. By using an LLM-based search, PlanKG's search procedure is probabilistic in contrast to search procedures known in database systems, e.g., SQL.

Activity sequences of selected plan descriptions are used as context for the generation of a plan (**set_actseq**). Similar to the k-nearest neighbors algorithm, the number k of selected plan descriptions is a hyperparameter of the generation procedure. The instantiation of the generation prompt pattern including the given context (**set_actseq**) and an optional sketch of the intended plan (*cm*) is input for the LLM-based generation of a plan description.

< Create **pm** with **cm** based on: ***set_actseq*** >

By setting the parameters of the generation prompt pattern, users of PlanKG can control the context for plan generation. As our evaluation studies indicate, activity sequences are sufficient representations for the task of plan generation over the more verbose full-text plan texts. In summary, the PlanKG method supports contextualized and controlled search and generation of full-text plan descriptions. The PlanKG method consists of a three-step transformer pipeline: (1) decoder-only transformer for knowledge graph extraction (Brown et al., 2020; Vaswani et al., 2017), (2) encoder-only transformer for knowledge graph embedding (Reimers and Gurevych, 2019), and (3) decoder-only transformer for plan generation. We have applied PlanKG only to sequential plans but will extend PlanKG to more complex ones.

3.5 Evaluation of search and generation results

Evaluation search results can be conducted based on the percentage of clustered context markers. For instance, movies are often associated with genres and production plans with the type of product. Therefore, we evaluate the accuracy of clusters by the separation percentage according to context markers. All PlanKG parameters can be tested in controlled studies: markers of search prompts and models for embeddings, dimensionality reduction, and clustering.

Evaluation of the novelty of generation results requires either domain expertise or simulation environments in which generated plans can be tested. Both are beyond the scope of this paper. Studies with domain experts are part of future work.

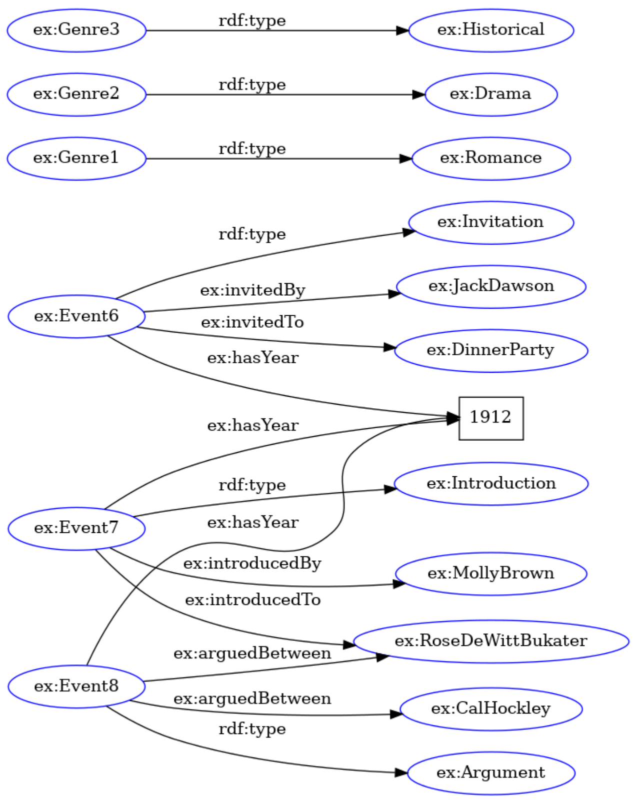

Figure 3. Knowledge graph excerpt for the movie Titanic

4 Study: Movie Plots

A plot is a sequence of events in which each event affects the next one through the principle of cause-and-effect (Forster, 1981; Prince, 2003). Hence, plots are sequential plans. We demonstrate the use of the PlanKG method for searching plots that are similar to a search prompt and even the generation of “novel" plots according to generation prompts. We show how PlanKG is used as a method for controlling the outcome of the search phase and generation phase as much as possible.

4.1 Datasets

To demonstrate the PlanKG method in a practical context and to obtain an initial validation of our work, we conducted a study using movie plots. The dataset consists of 80 movie plots from four different genres (action, comedy, horror, and romance) with 20 movies each (source: wikipedia.org). For each movie plot, summarizations were extracted by using chatGPT (version 3.5) with the prompt:

“Generate a short summary for the following movie plot: FTPD”

4.2 Knowledge Graph generation from movie plot descriptions

Knowledge graphs were extracted from plot descriptions by using an LLM (here: chatGPT) with an instantiation of the search prompt pattern. For instance:

“Format events and genre in RDF/N3 from: FTPD”

Figure 3 shows a visualization of the LLM-generated knowledge graph representation of the movie *Titanic*, with the blue ellipses indicating entities and the black arrows indicating relations. In this case, the entities (e.g., characters) that are related to each other through an event they are linked to are also depicted as such in the graph visualization.

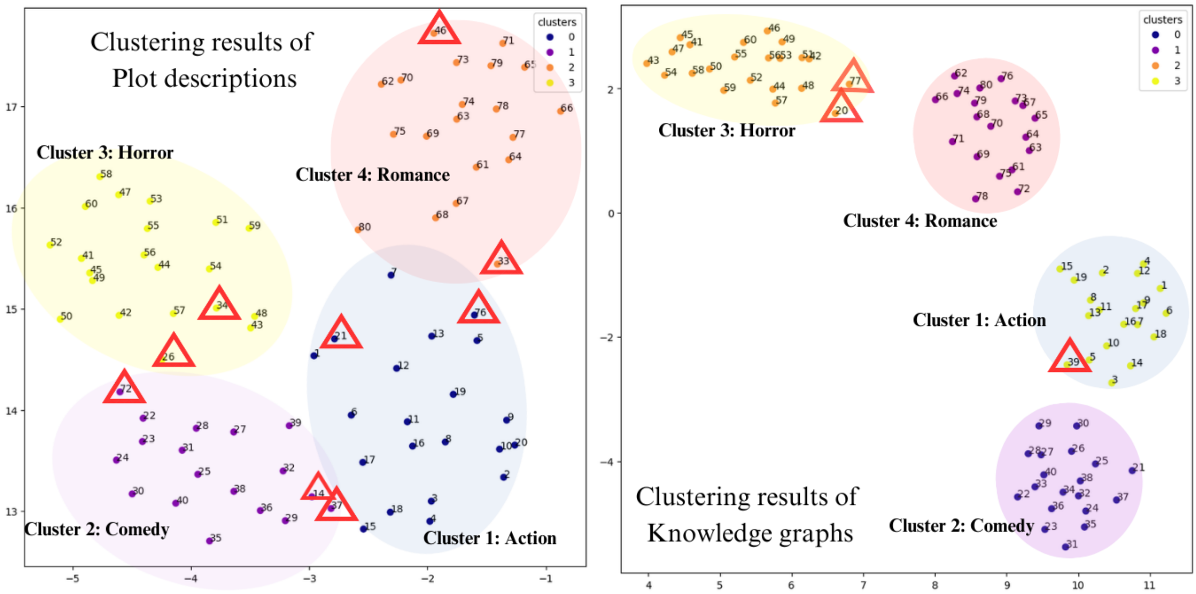

Figure 4. *Clustering results on activity sequences and genres. Points 1-20, 21-40, 41-60, and 61-80 represent action, comedy, horror, and romance movie activity sequences respectively. On the left: clustering obtained using the full-text plot descriptions from wikipedia.org. On the right: clustering obtained using knowledge graphs. Red triangles indicate misclassifications.*

Clusters from full-text descriptions are much more heterogeneous and less accurate (88\%) than clusters based on knowledge graphs (96\%) relative to the context marker ``genre'', as can be seen in the clustering visualizations in Figure 4. This can be interpreted as such as that the process of extracting knowledge graphs from the original text and transforming them into activity sequences led to more precise, meaningful embeddings (relative to the context marker).

4.3 Extraction of activity sequences

In the next step, we extracted the sequential plans for each of the movies from the knowledge representations, in the form of activity sequences from the movie plot and sets of genres as a comma-separated string format. These included all the relevant events from the plot (e.g., in the case of the movie *Titanic*, events: *TreasureHunting*, *Salvage*, etc.) and the genre (e.g., *Romance*, *Drama*, etc.). The resulting activity sequence is a highly compressed version of the FTPD having an average compression factor of 10 (Number of tokens in FTPD/Number of tokens in activity sequence).

4.4 Embeddings and clustering of activity sequences

This results in three sets of interest for comparison: the FTPD of movies from Wikipedia, sequential plan extracted from their knowledge graph representations, and the LLM-generated short summarizations of the plots.

We created the embeddings for all three items in the dataset using the SBERT embedding model with a vector length of 768. In order to perform non-linear dimensionality reduction and project the vectors into a two-dimensional space, we utilized Uniform Manifold Approximation and Projection (UMAP).

The cluster analysis was performed using k-Means clustering on the activity sequences extracted from the knowledge graphs, the plot summarization, and the raw plot descriptions. The number of clusters is set to 4 since there are 4 genres.

4.5 Evaluation of search and generation results

PlanKG			**Plot descriptions**			**Summarizations**		
Action	Come.	Horror	Action	Come.	Horror	Action	Come.	Horror
0.841	0.467	0.391	0.370	0.416	0.256	0.437	0.282	0.480
0.824	0.446	0.447	0.356	0.275	0.275	0.497	0.341	0.355
0.908	0.382	0.410	0.294	0.315	0.334	0.269	0.454	0.436
0.846	0.521	0.455	0.322	0.365	0.291	0.209	0.350	0.411
0.864	0.397	0.488	0.375	0.416	0.375	0.372	0.442	0.384
0.753	0.469	0.465	0.144	0.361	0.172	0.302	0.455	0.436
0.864	0.377	0.534	0.436	0.304	0.227	0.400	0.450	0.372
0.760	0.548	0.469	0.459	0.333	0.368	0.586	0.304	0.395
0.862	0.448	0.458	0.336	0.412	0.316	0.338	0.458	0.432
0.857	0.526	0.373	0.348	0.226	0.194	0.449	0.340	0.378
0.838	0.458	0.449	0.344	0.342	0.281	0.386	0.388	0.408

Table 1. Example cosine similarity scores of a reference action movie (title: \textit{Extraction}, 2020) to other action and other genre movies. The last row contains the average similarity values for each category. Category romance is not shown.

The clustering results are visualized in Figure 4 and show how close the embeddings of the respective representations of the movie plots are to each other in the embedding space. Although the activity sequences are a 10-times compressed version of the full-text descriptions, the clustering of the activity sequences from the knowledge graph representations leads to a significantly clearer distinction of the genres in the embedding space; i.e., the distances between the clusters are greater. From the clustering of activity sequences, almost all of the movie points are grouped into a similar cluster. However, clustering of the summarizations of plots as well as the full-text plot descriptions show several incorrect grouping.

To analyze this in a more detailed manner, we created a benchmark based on cosine similarity metrics, as shown in Table 2. We selected a test set for each of the genres in every dataset and found the similarity scores of these to all the genres in each set. Table \ref{tab1} displays the comparisons of an action movie (*Extraction*) as a reference to 10 other movies from each genre for each of the three datasets. Naturally, the similarity should be highest for those belonging to the action genre. In our study, we found that this was not consistently the case for either the plot descriptions or the LLM-generated summarizations. On other hand, we found that the use of knowledge graphs obtained by PlanKG strongly improved the genre matching by achieving significantly higher cosine similarity values between movies of the same genre than the previously mentioned resources, as can be seen in the last row of Table \ref{tab1}, which provides the respective average values for each combination of the three categories and genres. The mean similarity values from our PlanKG method for action movies is 0.838 whereas the mean from plot descriptions is 0.344. A paired t-test showed that this difference was significant ($t(9) = 17.38$, $p < 0.05$, confidence interval(95) = [0.42, 0.558153], Cohen's $d = 5.49$). Similar is the case for summarizations ($t(9) = 10.3014$, $p<0.05$, Confidence interval(95) = [0.352742, 0.551258], Cohen's $d = 3.25$). Thus, this gives proof to the hypothesis defined in RQ2. The activity sequences represented as knowledge graphs act as compressed representations of FTPD and similarity between the plans can be obtained efficiently by utilizing the embeddings of the knowledge graphs. Even with a compression factor of 10, the activity sequence can retain the information of the FTPD and gives better similarity search results than the full-text descriptions and summarizations.

4.6 Plan search and generation

The clustering technique and cosine similarity scores allow the users to search for sequential and full-text plans. Using PlanKG, the users can search a database of documents and obtain a set of activity sequences similar to the entered requirements (i.e., plan marker pm and context marker cm). The requirements are embedded in the activity sequence embedding space and the closest movie activity sequences are selected, i.e., the ones having the highest cosine similarity. The quality of the search thus relies on similar movies having high similarity values, and as shown previously, using activity sequences has been shown to deliver better results in this regard than textual plot descriptions.

Search prompt: Create movie plot with ``an intelligence agent during the cold war'' based on:
1. *Activity sequence from Insidious*: Haunted house occurrence, Paranormal activity recording, Visiting a psychic, Demonic manifestation, Ouija board encounter, Investigation of paranormal occurrences, ...\\
2. *Activity sequence from Annabelle*: Home invasion and murder, Paranormal activities, Birth of a child, Discovery of the doll and paranormal events, Investigation into the cult and demonic summoning, ...
3. *Activity sequence from Paranormal Activity*: Family moves into a new home, Encounter with a mysterious entity, Coma of one of the children, Frightening paranormal events in the house, Supernatural activity, ...
Requirements: Female lead character, former secret agent, wanted peaceful life, witness theft, etc.
Generated Result:
Title: Frozen Shadows
Plot description: Emily Sullivan, a skilled former intelligence agent haunted by her troubled past, seeks solace in a peaceful life in a quite town...

Table 1. Generation of a novel plot from activity sequences and requirements.

For a generation of a plan, these search results (*set_actseq*) are used for the instantiation of the *generation prompt pattern*. The context marker *cm* can be extended by additional information on activities and actors. Both, *set_actseq* and cm are requirements for plot descriptions. A new FTPD is generated using the instantiated generative prompt pattern.

For instance:

< Create movie plot with ``taking place in 1940 in Alaska and with a family of five as main characters based on: set_actseq>

(with set_actseq = {FTPD(Annabelle), FTPD(Insidious), FTPD(Paranormal Activity)})

The generated FTPD (cf. Table 3) satisfies all the requirements and is indeed taking into account the provided context. Also, the results are very detailed and precise. This shows that the controlled generation of novel FTPDs from a context of compressed activity sequences extracted from knowledge graphs and additional, specific requirements is feasible.

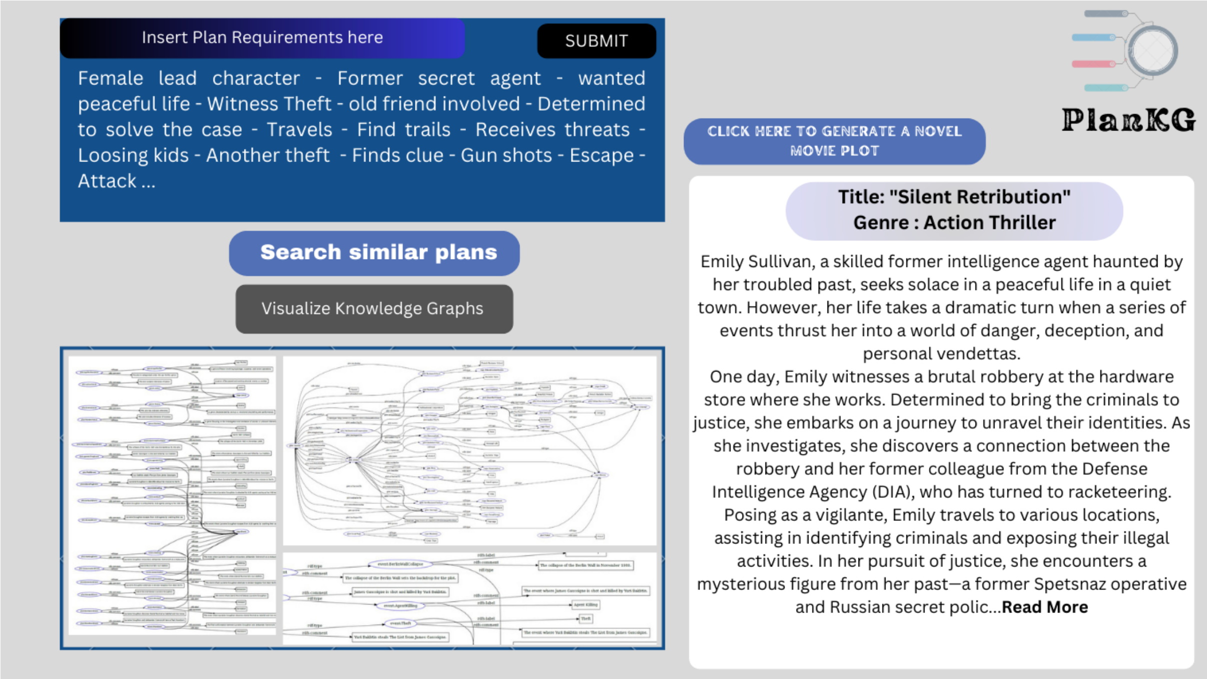

Figure 5. Use case of the PlanKG method in generating movie plots

5 Summary, limitations, and outlook

Recognizing the importance of plans in knowledge management and the possibility for them to be embedded in full-text documents, this research proposed the PlanKG method to search for sequential plans using knowledge graphs. Furthermore, PlanKG supports the generation of full-text plan descriptions based on plan requirements and selected activity sequences. To illustrate the method, it was applied to plots of movies because they can be considered sequential plans. The results indicate the usefulness of the method. This research has also shown the importance of knowledge graphs as intermediate knowledge structures.

Generalization of PlanKG is currently limited to domains in which extracting instance-based knowledge graphs with sequential activity sequences is sufficient. Future work focusses on abstract plan descriptions with parallel interrelated processes with loops, conditions, and constraints required for more complex domains, such as building construction or rescue situations. PlanKG is the starting point for this research.

Another limitation is that generated plan descriptions require evaluation in benchmarking experiments by scores similar to Bilingual Evaluation Understudy (BLEU) or Recall-Oriented Understudy for Gisting Evaluation (ROUGE). Currently, we focused on purely procedural knowledge as manifested in the activity sequences described in Section 3 while other types of knowledge, such as conditional or relational (Alavi and Leidner, 2001), may also potentially be processible by our method, which needs to be studied. We are currently working on expert studies in the domain of healthcare. In our future work, we will adapt PlanKG to the evolving capabilities of large language models (cf. (Huggingface, 2023)).

References

Ajzen, I. (1991). "The theory of planned behavior." Organizational behavior and human decision processes 50 (2), 179–211.

Alavi, M. and D. E. Leidner (2001). "Knowledge management and knowledge management systems: Conceptual foundations and research issues." MIS quarterly, 107–136.

Ale, M. A., C. M. Toledo, O. Chiotti, and M. R. Galli (2014). "A conceptual model and technological support for organizational knowledge management." Science of Computer Programming 95, 73–92.
Alexandrov, M., A. Gelbukh, and P. Rosso (2005). "An approach to clustering abstracts." In: Natural Language Processing and Information Systems: 10th International Conference on Applications of Natural Language to Information Systems, NLDB 2005, Alicante, Spain, June 15-17, 2005. Proceedings 10. Springer, pp. 275–285.
Borchard, A., D. L. Schwappach, A. Barbir, and P. Bezzola (2012). "A systematic review of the effectiveness, compliance, and critical factors for implementation of safety checklists in surgery." Annals of surgery 256 (6), 925–933.
Brown, T., B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. (2020). "Language models are few-shot learners." Advances in neural information processing systems 33, 1877–1901.
Chen, M., J. Tworek, H. Jun, Q. Yuan, H. P. d. O. Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman, et al. (2021). "Evaluating large language models trained on code." arXiv preprint arXiv:2107.03374.
Chen, P. P.- S. (1976). "The entity-relationship model—toward a unified view of data." ACM transactions on database systems (TODS) 1 (1), 9–36.
Chen, X., S. Jia, and Y. Xiang (2020). "A review: Knowledge reasoning over knowledge graph." Expert Systems with Applications 141, 112948.
Chowdhery, A., S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, et al. (2022). "Palm: Scaling language modeling with pathways." arXiv preprint arXiv:2204.02311.
Clarke, D. D., R. Forsyth, and R. Wright (1999). "Junction road accidents during cross-flow turns: a sequence analysis of police case files." Accident Analysis & Prevention 31 (1-2), 31–43.
Coyne, R. D. and J. S. Gero (1985). "Design knowledge and sequential plans." Environment and Planning B: Planning and Design 12 (4), 401–418.
Dalkir, K. (2017). Knowledge management in theory and practice. MIT press.
Devlin, J., M. -W. Chang, K. Lee, and K. Toutanova (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805.
Fikes, R. E. and N. J. Nilsson (1971). "STRIPS: A new approach to the application of theorem proving to problem solving." Artificial intelligence 2 (3-4), 189–208.
Forster, E. M. (1981). Aspects of the Novel, Edward Arnold, London, 1927.
Forstmann, B. U., E. -J. Wagenmakers, et al. (2015). An introduction to model-based cognitive neuroscience. Vol. 614. Springer.
Fu, B., Y. Qiu, C. Tang, Y. Li, H. Yu, and J. Sun (2020). A Survey on Complex Question Answering over Knowledge Base: Recent Advances and Challenges. arXiv: 2007.13069 [cs.CL].
Garrett, C. R., R. Chitnis, R. Holladay, B. Kim, T. Silver, L. P. Kaelbling, and T. Lozano-Pérez (2021). "Integrated task and motion planning." Annual review of control, robotics, and autonomous systems 4, 265–293.
Guo, Q., F. Zhuang, C. Qin, H. Zhu, X. Xie, H. Xiong, and Q. He (2020). A Survey on Knowledge Graph-Based Recommender Systems. arXiv: 2003.00911 [cs.IR].
Gutiérrez, C. and J. F. Sequeda (2021). "Knowledge graphs." Communications of the ACM 64 (3), 96–104.
Hochreiter, S. and J. Schmidhuber (1997). "Long short-term memory." Neural computation 9 (8), 1735–1780.
Hogan, A., E. Blomqvist, M. Cochez, C. d'Amato, G. d. Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, et al. (2021). "Knowledge graphs." ACM Computing Surveys (CSUR) 54 (4), 1–37.
Huang, W., P. Abbeel, D. Pathak, and I. Mordatch (2022). "Language models as zero-shot planners: Extracting actionable knowledge for embodied agents." In: International Conference on Machine Learning. PMLR, pp. 9118–9147.
Huggingface (2023). Transformers: Models. URL: https://huggingface.co/docs/%5C%5Ctransformers/main%5C_classes/model.

Jain, A. K., M. N. Murty, and P. J. Flynn (1999). "Data clustering: a review." ACM computing surveys (CSUR) 31 (3), 264–323.

Jia, W., M. Sun, J. Lian, and S. Hou (2022). "Feature dimensionality reduction: a review." Complex & Intelligent Systems 8 (3), 2663–2693.

Khorashadizadeh, H., N. Mihindukulasooriya, S. Tiwari, J. Groppe, and S. Groppe (2023). "Exploring In-Context Learning Capabilities of Foundation Models for Generating Knowledge Graphs from Text." arXiv preprint arXiv:2305.08804.

Koncel-Kedziorski, R., D. Bekal, Y. Luan, M. Lapata, and H. Hajishirzi (2022). Text Generation from Knowledge Graphs with Graph Transformers. arXiv: 1904.02342 [cs.CL].

LaValle, S. M. (2006). Planning algorithms. Cambridge university press.

LeCun, Y., Y. Bengio, and G. Hinton (2015). "Deep learning." nature 521 (7553), 436–444.

Lin, Z., M. Feng, C. N. d. Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio (2017). "A structured self-attentive sentence embedding." arXiv preprint arXiv:1703.03130.

Liu, V. and L. B. Chilton (2022). "Design guidelines for prompt engineering text-to-image generative models." In: Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems, pp. 1–23.

Maier, R. and T. Hadrich (2011). "Knowledge management systems." In: Encyclopedia of Knowledge Management, Second Edition. IGI Global, pp. 779–790.

Manola, F., E. Miller, B. McBride, et al. (2004). "RDF primer." W3C recommendation 10 (1-107), 6.

McInnes, L., J. Healy, N. Saul, and L. Großberger (2018). "UMAP: Uniform Manifold Approximation and Projection." Journal of Open Source Software 3 (29), 861.

McKee, R. (1997). "Story: Substance, structure, style and the principles of screenwriting. 1997." Kent, Great Britain: Methuen.

Mikolov, T., I. Sutskever, K. Chen, G. S. Corrado, and J. Dean (2013). "Distributed representations of words and phrases and their compositionality." Advances in neural information processing systems 26.

Miller, G., E. Galanter, and K. Pribram (1960). "Plans and the structure of behavior. Henry Holt"

Mineau, G. W., R. Missaoui, and R. Godinx (2000). "Conceptual modeling for data and knowledge management." Data & Knowledge Engineering 33 (2), 137–168.

Montes, A., H. Pacheco, H. Estrada, and O. Pastor (2008). "Conceptual model generation from requirements model: A natural language processing approach." In: Natural Language and Information Systems: 13th International Conference on Applications of Natural Language to Information Systems, NLDB 2008 London, UK, June 24-27, 2008 Proceedings 13. Springer, pp. 325–326.

Nickel, M., K. Murphy, V. Tresp, and E. Gabrilovich (2015). "A review of relational machine learning for knowledge graphs." Proceedings of the IEEE 104 (1), 11–33.

Olmo, A., S. Sreedharan, and S. Kambhampati (2021). "GPT3-to-plan: Extracting plans from text using GPT-3." arXiv preprint arXiv:2106.07131.

Paa., G. and S. Giesselbach (2023). "Pre-trained Language Models." In: Foundation Models for Natural Language Processing: Pre-trained Language Models Integrating Media. Cham: Springer International Publishing, pp. 19–78. ISBN: 978-3-031-23190-2. DOI: 10.1007/978-3-031-23190-2_2. URL: https://doi.org/10.1007/978-3-031-23190-2_2.

Parfitt, M. and V. Sanvido (1993). "Checklist of critical success factors for building projects." Journal of management in engineering 9 (3), 243–249.

Pilault, J., R. Li, S. Subramanian, and C. Pal (2020). "On extractive and abstractive neural document summarization with transformer language models." In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 9308–9319.

Prince, G. (2003). A dictionary of narratology. U of Nebraska Press.

Reimers, N. and I. Gurevych (Nov. 2019). "Sentence-BERT: Sentence Embeddings using Siamese BERTNetworks." In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics. URL: https://arxiv.org/abs/1908.10084.

Schank, R. C. and R. P. Abelson (1975). "Scripts, plans, and knowledge." In: IJCAI. Vol. 75. New York, pp. 151–157.

Schneider, P., T. Schopf, J. Vladika, M. Galkin, E. Simperl, and F. Matthes (2022). A Decade of KnowledgeGraphs in Natural Language Processing: A Survey. arXiv: 2210.00105 [cs.CL].

Simon, H. A. and A. Newell (1971). "Human problem solving: The state of the theory in 1970." American psychologist 26 (2), 145.

Singhal, A. et al. (2012). "Introducing the knowledge graph: things, not strings." Official google blog 5 (16), 3.

Trajanoska, M., R. Stojanov, and D. Trajanov (2023). "Enhancing Knowledge Graph Construction Using Large Language Models." arXiv preprint arXiv:2305.04676.

Turban, E. and P. R. Watkins (1986). "Integrating expert systems and decision support systems." Mis Quarterly, 121–136.

Valmeekam, K., A. Olmo, S. Sreedharan, and S. Kambhampati (2022). "Large Language Models Still Can't Plan (A Benchmark for LLMs on Planning and Reasoning about Change)." arXiv preprint arXiv:2206.10498.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin (2017). "Attention is all you need." Advances in neural information processing systems 30.

Walker, I., S. Reshamwalla, and I.Wilson (2012). "Surgical safety checklists: do they improve outcomes?" British journal of anaesthesia 109 (1), 47–54.

Wang, K., Z. Shen, C. Huang, C.-H. Wu, Y. Dong, and A. Kanakia (Feb. 2020). "Microsoft Academic Graph: When experts are not enough." Quantitative Science Studies 1 (1), 396–413. ISSN: 2641-3337.

Wang, Z., J. Zhang, J. Feng, and Z. Chen (2014). "Knowledge graph and text jointly embedding." In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1591–1601.

Xie, Y., C. Yu, T. Zhu, J. Bai, Z. Gong, and H. Soh (2023). "Translating natural language to planning goals with large-language models." arXiv preprint arXiv:2302.05128.

Yoon, Y., T. Guimaraes, and Q. O'Neal (1995). "Exploring the factors associated with expert systems success." MIS quarterly, 83–106.