MSc Data Science Dissertation Proposal

Leveraging Knowledge Graphs To Improve Large Language Models

Fasih Munir
fasih.munir@city.ac.uk
Supervisor: Tillman Weyde

1. Introduction

Large Language Models (LLM) have proven to be versatile, general purpose machines that are able to be applied to a broad range of tasks that range from text summarization, code assistance and creative work such as poetry writing. However, this broad applicability comes at the cost of being slow to train, needing extensive hardware and data and being expensive to train and operate (Naveed et al. 2023). Additionally, while LLMs are remarkable general purpose machines, their effectiveness is limited when challenged with tasks that require specific domain context such as diagnostic medicine (Ullah et al. 2024) as they are not directly trained on that data. These costs limit the extent to which a LLM can be adopted by consumers given that many consumers have specific needs which current LLMs may not be able to capture and creating their own LLM would be too costly.

Recently, Wei employed a technique called chain-of-thought (CoT) reasoning where the essential idea was for the LLM to receive prompt examples that were broken down into step by step reasoning (Wei et al. 2022). It was observed that this technique was able to broaden the ability of sufficiently large models (such as GPT-3) however limitations still exist as at times the reasoning chains may not entirely be correct for which more data would be required to fix.

Given the reasoning limitations and data requirements of LLMs, Knowledge Graphs (KG) have emerged as potential candidates that can augment the abilities of LLMs and can be grouped in 3 categories, LLM-augmented KGs, KG-enhanced LLMs and a combination of the augmentation and enhancement (Pan et al. 2023). Simply put, a KG is a graph-like structure that is built of objects and things that connect those objects (Fensel et al. 2020). For example, a KG is able to represent the idea that “a pizza has a deep pan base” (Jain et al. 2018) in the form of a triple (a structure of 3 things) where an object “pizza” is connected to an object “deep pan base” by a property, for example, “hasBase”. KGs are thus able to store knowledge and the context of that knowledge with well defined ontologies. These ontologies can be used to develop reasoning prompts, as seen by the creation of a chain-of-knowledge (CoK) (Wang et al. 2023) where the triple structure forms the base of the prompts.

1.1 Purpose and Objectives

In this paper, our purpose is to build on the idea of CoK and present an alternative solution. We will aim to answer the question:
Can we improve the quality of the output of a LLM by leveraging knowledge graphs through exploring minimal test cases that enable testing different architectures and algorithms.

The work of Wang et al. (2023) employed a generalized KG and made multiple calls to the graph for their reasoning prompts. Given that without domain specific context LLMs can struggle, we propose to use a domain specific KG, with a well structured ontology, that can be queried once to get all the related object properties. The KG we will use will be for the Dyck language which is an algebraic language with simple properties (Berstel 2007). This will be an atypical KG as the structure of Dyck words is stable with few complex relations between words such as nested words. Typical KGs have their strength in being able to capture deeper semantic relations, but the simplicity of our approach will be useful in providing a strong starting point to build from. The properties will be transformed into reasoning prompts which, an already available, LLM (such as GPT-3) will recursively ask itself and depending on either true or false, will continue or stop. This will add to the current body of research by focusing on proving or disproving the need for well constructed graphs, specific graphs for specific contexts and querying the graph once to obtain all the reasoning parameters for the LLM to prompt itself.

The following objectives and measures will outline what this research aims to accomplish:

1. Finalize a review on the current methods of enhancing LLMs with a KG

The output of this will be a thorough report on the progress of research in this domain. This will additionally help in finding other suitable evaluation metrics.

2. We will create an ontology for the Dyck language in Protege

To the best of our knowledge, no ontology formalizing the rules and grammar exists for the Dyck language. We will adhere to the principles laid out by Berstel (2007). We will then create multiple Dyck words and pass them to the reasoner in Protege and assess the assigned properties for validity. The use of this language, although not tied directly to an industry, can serve as a ground for extension into other domains such as music notation that has rules, for example, around chord structures.

3. Create a function in Python that is able to query the relevant object properties for a given LLM input and allow the pretrained LLM to recursively prompt itself

This will be considered successful if the pipeline is able to run end to end successfully with data being correctly outputted.

4. Assess different prompting use cases that include creation of words, checking the validity of words or understanding more about the language

This will be coupled with a detailed error analysis that will use examples of incorrect answers from pretrained models before employing the KG. These examples will further be tested on models enhanced by the KG to assess correctness.

Our review will enable a wider understanding of the Dyck language and its extensions, evaluation metrics and other models and domains to test on. This will lead to the creation of the first version of our ontology which in turn will be tested and evaluated in our python environment. During the process of testing we will aim to explore other architectures be it a different LLM or a different KG.

We expect this work to increase the adoption of LLMs for context specific industries such as music or medicine and further, organizations in those industries that may not have the resources to train or fine-tune their own LLM. For example, within music education, a LLM enhanced by a music theory KG would be more trusted to teach students about theory as it would need to adhere to the defined grammar within the crafted ontology. Although KGs can be difficult to create, the domain knowledge required will already be available within organizations. With training, the process of KG creation can be distributed amongst domain experts, the cost of which would be much lower compared training and maintaining a LLM.

2. Critical Context

2.1 Overview of KG LLM Combinations

Pan et al. (2023) provides a detailed overview about the different ways KGs and LLMs have been used together. Firstly, KGs can be used to improve LLMs by either being used as additional data during the training process or directly affecting the output to ensure factual accuracy. Secondly, LLMs can use their ability to parse and analyze large bodies of unstructured text to extract information that can be used to expand KGs and keep them up to date. Thirdly, both can be used in a pipeline to enhance each other where the insights from an LLM can populate a KG which can then continue providing richer context either during re-training or at the output.

Many ways to enhance KGs with LLMs have been proposed. Zhang et al. (2020) found that knowledge representations in large KGs can have degraded performance due to noisy data annotations and so they propose to use the knowledge from a pretrained language model such as BERT to enhance entity and relation representation. Xie et al. (2022) also focused on completing KGs by using pretrained language models to complete missing triples within the graph. On the other hand, Hao et al. (2023) proposed to use a modification of the BERT model to create a KG end to end using the embeddings within the transformer model rather than using the knowledge to complete existing KGs. Wang et al. (2019) further proposed a model, KEPLER, where they use a LLM to encode entity descriptions and then simultaneously optimize the knowledge graph embeddings. While these studies are informative and have made significant advancements in problem solving such as being able to work with sparse examples (Zhang et al. 2020), all still rely on large transformer models that have compute limitations to either retrain or finetune, lack generalizability to specific contexts and have not been tested on specific contexts. These studies present valuable foundations upon which we can build our approach given we are aiming to affect only the output of an LLM which in turn should reduce the time and financial resources spent on large models.

Taking a look at how KGs enhance LLMs, this can broadly happen in 2 ways. The first is by using KGs as part of the training set for LLMs which would provide them with a deeper semantic understanding. Depending on the KGs used, these can be very useful for domain specific work. For example, Zhang et al. (2020) introduce E-BERT for e-commerce which is a LLM that, in its training phase, employs a domain specific product association knowledge graph. Their incorporation of the

product KG was able to improve product classification and review question answering. However they specifically noted the difficulty in creating the product KG due to the noise (they scraped Amazon product reviews) and additionally, once again, a LLM was being trained.

In this paper, we would like to shift the focus to the second way, how KGs enhance LLMs in the way they affect the output. This method provides a layer of fact checking and context that does not come at the cost of extensive retraining. KGs can still be difficult to create due to domain knowledge and process. However, domain knowledge is less troublesome as this should be available within an industry for example medicine which leaves only the process which, as we have seen, is an open area of research. Although a manual process of creation is still possible. We will make the case for domain specific KGs that can in turn continue research in automated KG creation.

2.2. KG-Enhanced LLM Output

Before studying how KGs enhance the output of a LLM, it is important to learn how we arrived at this point. We begin by understanding the role prompting plays within LLMs. CoT prompting is a technique that significantly improves the complex reasoning performance of LLMs and can emerge naturally in “sufficiently large” models (Wei et al. 2022). The researchers were able to achieve this by simply providing the LLM with a handful of reasoning examples. These examples were manually configured and were directly able to affect the output. For example, rather than asking the model to solve a math equation, they would instead lay out the equation step by step which then allowed the model to arrive at the correct answer. Kojima et al. (2022), took a different approach to CoT, where instead of providing reasoning examples, they simply added the phrase “let’s think step by step…” before each prompt. They were able to show dramatic accuracy gains on multiple benchmarks proving that simpler reasoning prompts should be explored before crafting examples or even finetuning models. Zhang et al. (2022) present AutoCoT, which builds on the usual CoT by automatically creating reasoning examples versus manually crafting them. Although they note that having diverse data is better for creating prompts with less mistakes. This brings us to Wang et al. (2023), who introduce CoK. They aimed to improve reasoning rationale by making the LLM produce evidence triples and explanation hints that come from their chosen, general, knowledge bases such as Wikidata. The triples are used to guide the LLM with factual evidence to make the reasoning chain reliable.

In this paper we will draw inspiration from CoT and CoK and present an alternative way for LLMs to reason reliably. Wang et al. (2023) noted a limitation of having finite triples. Given that they used a large and available set, and the nature of how knowledge grows, it is likely that there may never be a complete set of triples. It should further be noted that they invited domain experts to manually annotate triples to improve their reliability, highlighting the need for domain expertise. Thus, we will explore the use of a set of domain specific triples as reasoning chains. We will explore how difficult it would be to create these triples and whether they can be reliably used to improve the LLM output.

2.3 The Dyck Language

Our domain of choice, to start with, will be the Dyck language. Given that it is purely an algebraic language that consists of parentheses with simple rules such as having balanced parentheses (Berstel 2007), it will be relatively quick to build domain expertise as opposed to using a domain such as music or medicine where building the domain expertise takes years of training and finding the experts to help build an ontology can be a full research in itself. Dyck words also provide an area of sparse knowledge for which there is room to contribute with ontology creation and test the limits of LLMs with sparsity. Once we exhaust the possible use cases of this ontology, we may explore a different domain such as creating a small ontology for music theory if time permits.

3. Approaches

While a review of relevant papers has been conducted for preparing the critical context, an additional comprehensive review will be taken to establish best practices for ontology creation, to learn more about the Dyck language and its variations and assess which pretrained LLMs will work best for our research keeping in mind cost and access. We will also look at other possible avenues to assess our research outside of the ones outlined in section 3.3. Research will be conducted using online resources such as ACL Anthology and the City, University of London library. Further we will try to find other domain specific examples and compare results qualitatively.

3.1 Dyck Language Ontology

Currently, we are not aware of an ontology created for this language and will work to create it using the Protege software that is commonly used for ontologies within the KG domain. Given the relative simplicity of the language, this will form the foundation of our experimentation that can enable potential extension to other domains. Within the software we will work with Web Ontology Language (OWL) which is designed to represent knowledge and the relations between different kinds of knowledge. We will work through the explanations of Berstel (2007) and any other useful papers that we find during the comprehensive literature review and express the Dyck language using the OWL framework (Berstel's work has been used to express the thoughts around the language in this paper). This will include the creation of classes, properties and instances. Given a well defined OWL format, we will be able to then use the reasoning capabilities that are inherently present in the framework to assess the validity of Dyck words which will be useful when we are testing the accuracy of our ontology. Ontologies can be saved in various formats and can be exported to Python where we can use the ontology with RDFlib, which is a Python package designed to work with the triple structure of ontologies and graph representations. This will take our created ontology and store it in the Resource Description Framework (RDF) format. Having the ability to work with the ontology in Python will further allow us to query the graph structure using SPARQL Protocol and RDF Query Language (SPARQL) which is the language used to extract information from RDF formats. Technically it is very similar to the Structured Query Language (SQL) used to query traditional relational databases.

We will try to model the potential classes, object properties and some individuals which will inherently model the grammar of the language in some basic rules. The properties and classes will be made in the style of ontologies, for example ‘locatedIn’ where the first word is lowercase, there is no space and the second word is propercase (other variations exist but this is the general format). To use these properties and classes as valid LLM inputs, some simple string manipulation will be employed to turn ‘locatedIn’ into ‘located in’.This transformation is a more common string that is used for LLM prompts and given that LLMs predict the sequence of tokens, the transformed characters are likely more useful tokens to use.

3.1.1 Potential Classes

Since the language is built on parentheses some immediate classes that can be created are for left and right parentheses. Additionally, the language can extend to other brackets and so we can include left and right curly or square brackets as well. These classes could be called ‘Characters’ with subclasses of ‘LeftParentheses’ and ‘RightParentheses’. These are also known as opening or closing symbols, which can also be possible classes.

3.1.2 Potential Object Properties

Dyck words are supposed to start and end with parentheses and these parentheses need to be balanced. This is to say that a “()” is a Dyck word but “)(“ is not a Dyck word. The reason why the second word is not a Dyck word is because the opening and closing parentheses are not balanced by their respective opening and closings. Thus, possible object properties can include ‘startsWith’ and ‘endsWith’ which relate to the opening and closing symbols respectively. Given that our ontology will also likely use brackets, other object properties such as ‘contains’ can be used to signify that Dyck words should only contain the given characters as defined by the classes. Additionally, to check for balance, a rule such as ‘isEqualTo’ could be created for correct reasoning when transforming the object property to a string input for the LLM.

3.2 Python Prompt Function and Model

Since we are trying to augment already created models, we will not be training our own. This leads us to using pretrained models such as GPT-3 which has also been used in the papers discussed in our critical context. GPT-3 is easily accessible with an OpenAI account and the free credit can be used to run small scale tests. Since it is API accessible, we will be able to work with the model in Python and have our ontology and model in the same environment. While GPT-3 is one possible model (and a useful one for the sake of comparison), our literature review will conduct a further analysis of the possible models that we can employ that will balance cost and access. Some newer, open source models include Llama and Falcon that have the advantage of being run locally. The prompt function itself is to be designed but will likely involve the use of SPARQL and string transformations. The answer of the transformed SPARQL output will be used to then get the LLM to recursively prompt itself. This will emulate reliable reasoning. For example, when asking if a word is valid, the query may pull the related object properties for valid words, the object properties get transformed into useful

string inputs which are then used as reasoning prompts for the LLM to assess the answer. Additionally, since LLM inputs can vary, there is also the possibility of converting the string input to a LLM directly to SPARQL using a tool such as AutoSPARQL. We will test with various string inputs including (but not limited to) “Is…a Dyck word?”, “What are the properties of Dyck words?” and “Can you help me write a Dyck word that has a length of 21 characters”.

3.3 Evaluation

Our evaluation will be two fold. The first will be a detailed error analysis using prompts that give incorrect responses on current models. For example, when prompting GPT-4 to define the rules of Dyck words it provided the output “Invalid Dyck Word: *XXYYXY* or (())()”. This is an incorrect output as the word it generated is in fact a valid Dyck word as it consists of parentheses that are correctly balanced. Another example is when when prompting GPT3.5 to identify if “()()()()()()()()()()()()()()(())((((()))))()()()()()()()()()()((((()))))(())(())(())()()()()(())(())()()()()(())(())()()()()()()()()()())(())(())(())(())((((((()))))))()()()()()()()()” was a Dyck word, it was able to say that it was not because it lacked overall balance. However, when prompted to find where it went wrong, it was not able to and explained that “Towards the end, there are multiple consecutive closing parentheses without their corresponding opening parentheses: "))))"”. This is also incorrect as can be seen in the example. Other such examples will be generated and then compared on our chosen models such as GPT-3. Our model will first be prompted on its own and then prompted with the KG in use and the outputs will be qualitatively assessed.

The second way we will assess our work is by using the same prompts that gave incorrect results but applying the techniques in the papers referenced in our critical context. Specifically, we will test with the CoT examples and single line (or zero shot) prompt of “think step by step…”. We will look to see how well the other techniques perform in a domain specific context and how much better or worse our technique does.

If our extended literature review results in a performance metric that can be applied to our use case, that may be employed as well.

4. Work Plan

Figure 1 (right), summarizes how the work plan will flow, followed by Figure 2 (our Gantt chart below) that gives a detailed breakdown of the tasks and milestones. The project officially begins on the 1st of July 2024 and will continue for a period of 3 months. We plan to complete

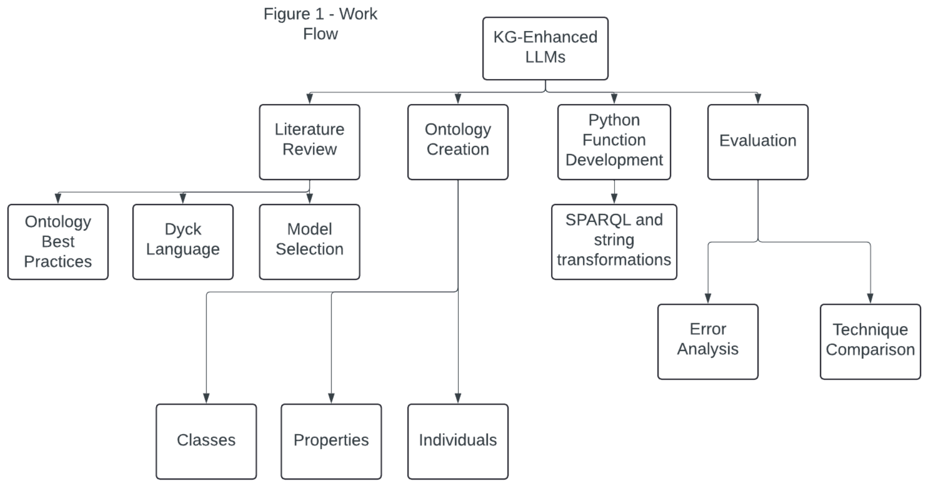
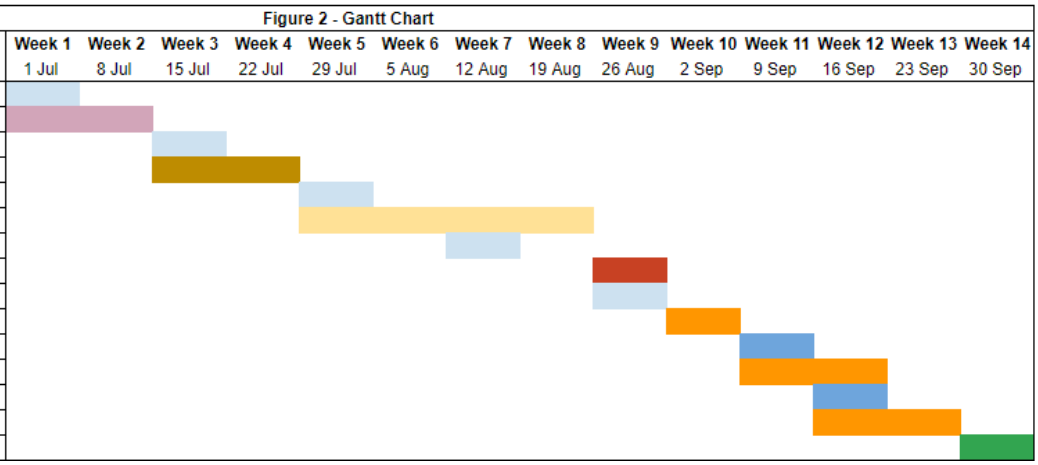


Figure 1 - Work Flow

the final draft within Week 13 which gives an approximate 1 week buffer in case of unforeseen issues (Submission due 2nd October 2024). Bi-weekly meetings with the supervisor have been scheduled but can be ignored depending on progress. Major milestones include creating the ontology (Week 4), creating the testing environment in python (Week 5), completing evaluation (Week 9) and second draft review (Week 12).

Figure 2 - Gantt Chart

Task	Start Date	End Date
Kickoff Meeting	1 Jul	1 Jul
Literature Review	1 Jul	15 Jul
Progress Meeting	15 Jul	15 Jul
Ontology Creation	15 Jul	29 Jul
Progress Meeting	29 Jul	29 Jul
Python Code	29 Jul	26 Aug
Progress Meeting	12 Aug	12 Aug
Research Evaluation	26 Aug	2 Sep
Progress Meeting	26 Aug	26 Aug
First Draft	2 Sep	9 Sep
First Draft Review	10 Sep	10 Sep
Second Draft	11 Sep	18 Sep
Second Draft Review	19 Sep	19 Sep
Final Draft	20 Sep	27 Sep
Submit Report	2 Oct	-



5. Risks

The following risks have been identified after reading through the resource framework suggested by Dawson (2009). There are no external participants.

Risk	Likelihood (1 - 5)	Consequence (1 - 5)	Impact (LxC)	Mitigation
Unforeseen Illness	2	4	8	1 week buffer, extension application due to Extenuating Circumstances
Incorrect Time Estimation	2	4	8	1 week buffer, adapt scope of project if necessary.
Semantic Web Technology Requiring More Expertise	3	5	15	Thorough paper review and contact with Semantic Web Professor
API Limitations For Accessing Models	1	3	3	Pay subscription, use open source models on City machines.
Hardware Failure/Hardware Loss	1	5	5	Cloud backups through Google Colab and Google Workspace
Work/Data Getting Lost	1	5	5	Cloud backups through Google Colab and Google Workspace

6. References

1. Naveed, H., Khan, A.U., Qiu, S., Saqib, M., Anwar, S., Usman, M., Akhtar, N., Barnes, N. and Mian, A. (2023). A Comprehensive Overview of Large Language Models. [online] arXiv.org. doi:https://doi.org/10.48550/arXiv.2307.06435.
2. Ullah, E., Anil Parwani, Mirza Mansoor Baig and Singh, R. (2024). Challenges and barriers of using large language models (LLM) such as ChatGPT for diagnostic medicine with a focus on digital pathology – a recent scoping review. Diagnostic Pathology, 19(1). doi:https://doi.org/10.1186/s13000-024-01464-7.
3. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q. and Zhou, D. (2022). Chain of Thought Prompting Elicits Reasoning in Large Language Models. arXiv:2201.11903 [cs]. [online] Available at: https://arxiv.org/abs/2201.11903.
4. Fensel, D., Şimşek, U., Angele, K., Huaman, E., Kärle, E., Panasiuk, O., Toma, I., Umbrich, J. and Wahler, A. (2020). Introduction: What Is a Knowledge Graph? Knowledge Graphs, pp.1–10. doi:https://doi.org/10.1007/978-3-030-37439-6_1.
5. S. Jain, R. Gupta and R. K. Dwivedi, "Generating Patterns from Pizza Ontology using Protégé and Weka Tool," 2018 International Conference on System Modeling & Advancement in Research Trends (SMART), Moradabad, India, 2018, pp. 126-131, doi: 10.1109/SYSMART.2018.8746935.
6. Pan, S., Luo, L., Wang, Y., Chen, C., Wang, J. and Wu, X. (2023). Unifying Large Language Models and Knowledge Graphs: A Roadmap. [online] arXiv.org. doi:https://doi.org/10.48550/arXiv.2306.08302.
7. Wang, J., Sun, Q., Chen, N., Li, X. and Gao, M. (2023). Boosting Language Models Reasoning with Chain-of-Knowledge Prompting. [online] arXiv.org. doi:https://doi.org/10.48550/arXiv.2306.06427.
8. Jean Berstel Transductions and Context-Free Languages. (2007). Available at: https://www-igm.univ-mlv.fr/~berstel/LivreTransductions/LivreTransductions.pdf
9. Zhang, Z., Liu, X., Zhang, Y., Su, Q., Sun, X. and He, B. (2020). Pretrain-KGE: Learning Knowledge Representation from Pretrained Language Models. doi:https://doi.org/10.18653/v1/2020.findings-emnlp.25.
10. Xie, X., Zhang, N., Li, Z., Deng, S., Chen, H., Xiong, F., Chen, M. and Chen, H. (2022). From Discrimination to Generation: Knowledge Graph Completion with Generative Transformer. [online] arXiv.org. doi:https://doi.org/10.1145/3487553.3524238.
11. Hao, S., Tan, B., Tang, K., Ni, B., Shao, X., Zhang, H., Xing, E.P. and Hu, Z. (2023). BertNet: Harvesting Knowledge Graphs with Arbitrary Relations from Pretrained Language Models. [online] arXiv.org. doi:https://doi.org/10.48550/arXiv.2206.14268.
12. Wang, X., Gao, T., Zhu, Z., Zhang, Z., Liu, Z., Li, J. and Tang, J. (2019). KEPLER: A Unified Model for Knowledge Embedding and Pre-trained Language Representation. arXiv (Cornell University). doi:https://doi.org/10.48550/arxiv.1911.06136.
13. D. Zhang et al., “E-BERT: A Phrase and Product Knowledge Enhanced Language Model for E-commerce,” Sep. 2020, doi: https://doi.org/10.48550/arxiv.2009.02835.
14. Kojima, T., Gu, S.S., Reid, M., Matsuo, Y. and Iwasawa, Y. (2022). Large Language Models are Zero-Shot Reasoners. arXiv:2205.11916 [cs]. Available at: https://arxiv.org/abs/2205.11916.
15. Zhang, Z., Zhang, A., Li, M. and Smola, A. (2022). Automatic Chain of Thought Prompting in Large Language Models. arXiv:2210.03493 [cs]. Available at: https://arxiv.org/abs/2210.03493.
16. Dawson, C. (2009). Projects in Computing and Information Systems A Student’s Guide. [online] Available at:https://repository.dinus.ac.id/docs/ajar/Projects-in-Computing-and-Information-Systems-A-Student%e2%80%99s-Guide-2nd-Edition-April-2009.pdf.

Research Ethics, Part A: Ethics Checklist

A.1 If you answer YES to any of the questions in this block, you must apply to an appropriate external ethics committee for approval and log this approval as an External Application through Research Ethics Online - https://ethics.city.ac.uk/		*Delete as appropriate*
1.1	Does your research require approval from the National Research Ethics Service (NRES)?	**NO**
1.2	Will you recruit participants who fall under the auspices of the Mental Capacity Act?	**NO**
1.3	Will you recruit any participants who are currently under the auspices of the Criminal Justice System, for example, but not limited to, people on remand, prisoners and those on probation?	**NO**
A.2 If you answer YES to any of the questions in this block, then unless you are applying to an external ethics committee, you must apply for approval from the Senate Research Ethics Committee (SREC) through Research Ethics Online - https://ethics.city.ac.uk/		*Delete as appropriate*
2.1	Does your research involve participants who are unable to give informed consent?	**NO**
2.2	Is there a risk that your research might lead to disclosures from participants concerning their involvement in illegal activities?	**NO**
2.3	Is there a risk that obscene and or illegal material may need to be accessed for your research study (including online content and other material)?	**NO**
2.4	Does your project involve participants disclosing information about special category or sensitive subjects?	**NO**
2.5	Does your research involve you travelling to another country outside of the UK, where the Foreign & Commonwealth Office has issued a travel warning that affects the area in which you will study?	**NO**
2.6	Does your research involve invasive or intrusive procedures?	**NO**
2.7	Does your research involve animals?	**NO**

2.8	Does your research involve the administration of drugs, placebos or other substances to study participants?	**NO**
A.3 If you answer YES to any of the questions in this block, then unless you are applying to an external ethics committee or the SREC, you must apply for approval from the Computer Science Research Ethics Committee (CSREC) through Research Ethics Online - https://ethics.city.ac.uk/ **Depending on the level of risk associated with your application, it may be referred to the Senate Research Ethics Committee.**		*Delete as appropriate*
3.1	Does your research involve participants who are under the age of 18?	**NO**
3.2	Does your research involve adults who are vulnerable because of their social, psychological or medical circumstances (vulnerable adults)?	**NO**
3.3	Are participants recruited because they are staff or students of City, University of London?	**NO**
3.4	Does your research involve intentional deception of participants?	**NO**
3.5	Does your research involve participants taking part without their informed consent?	**NO**
3.6	Is the risk posed to participants greater than that in normal working life?	**NO**
3.7	Is the risk posed to you, the researcher(s), greater than that in normal working life?	**NO**
A.4 If you answer YES to the following question and your answers to all other questions in sections A1, A2 and A3 are NO, then your project is deemed to be of MINIMAL RISK. **If this is the case, then you can apply for approval through your supervisor under PROPORTIONATE REVIEW. You do so by completing PART B of this form.** **If you have answered NO to all questions on this form, then your project does not require ethical approval. You should submit and retain this form as evidence of this.**		*Delete as appropriate*
4	Does your project involve human participants or their identifiable personal data?	**NO**