國立臺灣大學文學院語言學研究所

碩士論文

Graduate Institute of Linguistics

College of Liberal Arts

National Taiwan University

Master's Thesis

基於大型語言模型的外語報刊數據意見挖掘

Exploring LLM-Based Opinion Mining in Foreign-Language Newspaper Data

戴宓

Deborah Watty

指導教授: 謝舒凱 博士

Advisor: Shu-Kai Hsieh, Ph.D.

中華民國 114 年 7 月

July, 2025

doi:10.6342/NTU202502354

Acknowledgments

I am grateful for the many kinds of support I received throughout this project, whether academic, practical, or personal. What follows is by no means an exhaustive list of those who supported me.

First and foremost, I would like to thank my supervisor, Professor Shu-Kai Hsieh, for his guidance throughout the project.

I am also thankful to my examination committee members, Professor Chih-Sung Teng and Professor Yu-Yun Chang, for their time and constructive feedback. A special thanks to Professor Teng, who provided me with the People's Daily dataset early on in the project and sparked the idea for the sentiment analysis part of the study.

I would like to thank Egil Rønningstad for sharing reference code that greatly helped with the sentiment analysis experiment.

Thank you to Diffbot for providing free access to their API, and particularly to Leann Chen for going above and beyond in answering my questions about Diffbot during her free time.

Big thanks to everyone in my lab for sitting through a year's worth of progress reports and providing helpful suggestions. Special shoutout to Richard Lian for answering all my

doi:10.6342/NTU202502354

technical questions and helping with server access.

I am also grateful to everyone who took the time to fill out my long questionnaire. Their responses were key to building the test set for the sentiment analysis experiment.

Finally, I am especially grateful to my husband, Jens Niklas, for supporting me throughout the entire process, from discussing initial ideas and sitting through dependency hell with me late at night to being my audience for every practice run of my oral defense. Thank you.

doi:10.6342/NTU202502354

摘要

在日益緊密相連的世界中，跨越語言邊界理解事件如何被報導是一項重要的挑戰。本論文以《人民日報》（中國共產黨的官方報章）為例，探討大型語言模型（LLMs）對多語言意見挖掘之使用。雖然基於 LLM 的方法在近年來已被證實對意見挖掘任務非常有效，但其在多語言任務應用方面的研究仍然相對有限。

本論文的目標是評估基於 LLM 的方法在單語和多語語境下進行意見挖掘與問答的表現，並檢測內文與提示語言不相符是否會影響結果。第一個實驗聚焦於使用零樣本提示詞識別實體層級的情感，並比較中文、英文及德文的提示詞結果表現。然而，儘管情感分析提供了重要的洞察，卻無法提供關於文本內容的有效資訊。為了彌補此一缺陷，第二個探索實驗使用檢索增強生成（RAG）進行問答，並比較了三種不同架構在不同問題類型上的表現。

在這兩個實驗中，多語言 LLM（如 GPT-4、Gemini）均表現出穩健的性能，即使在數據資料與查詢語言不一致的情況下，表現差異也非常微小。零樣本提示在情感分析中展現了強大的潛力，對「日本」一詞的可視化情感分析反映了中日關係在一些重要事件期間的預期變化。而在問答任務中，RAG 架構的選擇則對性能表現影響顯著，不同架構在處理不同類型的問題時各有所長，突顯了根據任務調整方法的必要性。

這些研究結果表明了基於 LLM 的方法在多語言任務中的靈活性，即便數據

doi:10.6342/NTU202502354

資料和查詢語言有所不同，仍能為情感分析和問答任務提供有效的解決方案。

關鍵字：大型語言模型，多語言情緒分析，檢索增強生成，知識圖譜

Abstract

Understanding how events are reported across linguistic boundaries is a significant challenge in an increasingly interconnected world. This thesis explores the use of large language models (LLMs) for multilingual opinion mining, using The People's Daily, the official newspaper of the Communist Party of China, as a sample use case. While LLM-based methods have proven to be highly effective for opinion mining tasks in recent years, there is still relatively little research on their application to multilingual tasks.

The overall goal of this thesis was to assess the performance of LLM-based methods for opinion mining and question answering in both monolingual and multilingual contexts, evaluating whether mismatches between content and prompt languages impact outcomes. The first experiment focused on identifying entity-level sentiment using zero-shot prompting, comparing the performance of Chinese, English and German prompts. Although sentiment analysis provides valuable insights, it offers no information about the content of the texts. To address this gap, the second experiment explored question answering using

doi:10.6342/NTU202502354

Retrieval-Augmented Generation (RAG), comparing the performance of three different architectures across different question types.

Across both experiments, multilingual LLMs, such as GPT-4 and Gemini, showed robust performance, with minimal differences observed when data and query languages did not match. Zero-shot prompting demonstrated strong potential for sentiment analysis, with visualizations of sentiment toward Japan revealing expected shifts during key events in Sino-Japanese relations. For question answering, the choice of RAG architecture significantly influenced performance, with different architectures excelling at different types of questions, underscoring the need to tailor the approach to the task.

These findings underscore the versatility of LLM-based methods for multilingual tasks, offering effective solutions for sentiment analysis and question answering, even in cases where data and queries are in different languages.

Keywords: LLMs, Multilingual Sentiment Analysis, RAG, Knowledge Graphs

Contents

doi:10.6342/NTU202502354

doi:10.6342/NTU202502354

doi:10.6342/NTU202502354

doi:10.6342/NTU202502354

List of Figures

doi:10.6342/NTU202502354

List of Tables

doi:10.6342/NTU202502354

doi:10.6342/NTU202502354

1 Introduction

1.1 Motivation

In a globalized world, language barriers still pose challenges for accessing news from foreign countries. Following international developments often means relying on translations or reports filtered through the lens of local media. Reports in one's own country about foreign events tend to be both written from an outside perspective and shaped by relevance to the local audience, potentially overlooking context critical to the original culture. Official English-language versions of foreign publications may also be abridged, further limiting complete insight.

Access to foreign-language sources has greatly improved in recent years, thanks to advances in language technology. However, challenges remain, especially when it comes to capturing nuance and context. Idioms often lack direct equivalents in other languages, and political slogans can lose their connotations in translation. For example, the phrases "American Dream" and "中国梦" (China Dream) use similar wording but serve different rhetorical purposes in each culture (Callahan, 2017). Additionally, identifying relevant articles within vast foreign-language data and accurately assessing the political stance of a publication remain challenging. Large Language Models (LLMs) and Retrieval-Augmented Generation (RAG) have introduced new possibilities, such as aggregating

doi:10.6342/NTU202502354

information across multiple articles and explaining nuanced terms or culturally specific concepts. However, current research on opinion mining with LLMs has focused on English use cases, leaving a gap in best practices for applications to multilingual scenarios.

1.2 Dataset

To address this gap, this thesis uses the *People's Daily*, the official newspaper of the Communist Party of China, as a case study. Published since the 1940s, it has been the subject of many studies regarding its format, content, and use of language. As such, it is particularly suitable for experiments on opinion mining, as the plausibility of the results can be evaluated against the clearly established positions of the Chinese government as well as results of previous studies on the same dataset. The dataset[1] spans from May 1946 to September 2023, covering thousands of articles annually. The number of articles published each year is shown in Figure 1.1.

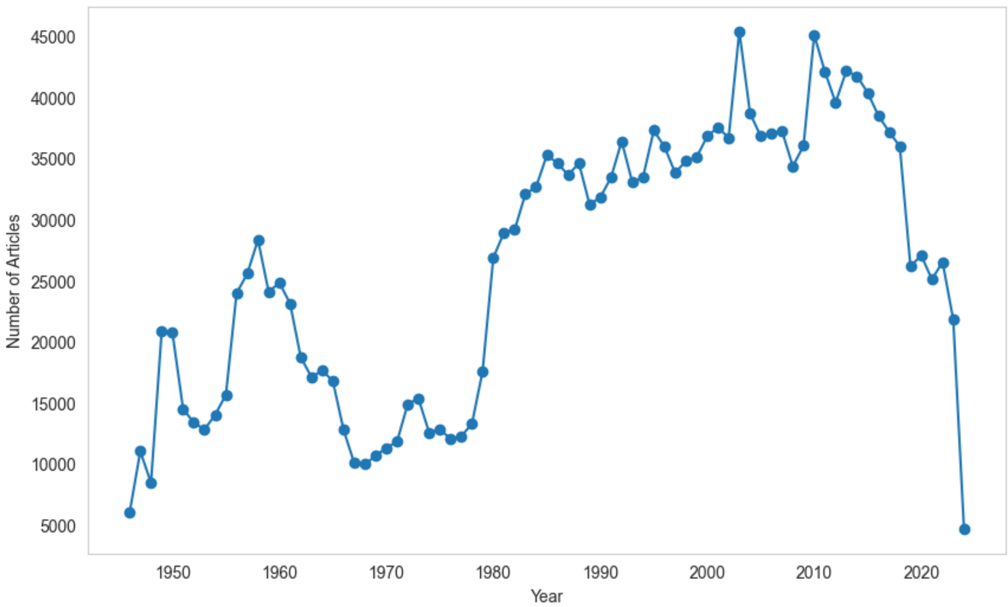


Figure 1.1: Number of articles per year in the People's Daily dataset.

[1] The dataset was retrieved from https://csteng.cc/docs/rmrb (accessed October 9, 2023).

doi:10.6342/NTU202502354

1.3 Goals and Research Questions

The goal of this thesis is to explore LLM-based methods for multilingual opinion mining on the People's Daily dataset. A central focus is determining whether mismatches between content and prompt languages impact outcomes.

Specifically, two main experiments are conducted:

- Sentiment Analysis: **Can zero-shot prompting effectively identify entity-level sentiment, and does the prompt language influence performance?**

  Previous research has used machine learning for sentiment analysis on People's Daily. This experiment evaluates whether zero-shot prompting can achieve comparable results and whether the language of the prompt (Chinese, English, or German) affects results.

- Question Answering: **How do different RAG architectures perform on diverse types of questions, and can they effectively handle queries in a language different from that of the datastore?**

  While sentiment analysis offers insights into the overall stance of texts, it does not uncover the actual content or specific details. To address this limitation, the second experiment investigates question answering by testing how three different RAG architectures perform across three different question types and assessing their ability to handle mismatches between the languages of the datastore and user query.

doi:10.6342/NTU202502354

Although sentiment analysis and question answering are traditionally treated as distinct tasks, they are united here by a shared goal: extracting meaningful information from newspaper data by prompting an LLM. This thesis explores whether prompt-based methods can serve as a flexible approach to analyzing foreign-language content across different information needs.

1.4 Structure of the Thesis

The remainder of this thesis is structured as follows: Chapter 2 provides a review of relevant literature, including previous work on the People's Daily and recent studies on LLM-based opinion mining. Chapter 3 presents the results of the sentiment analysis experiments, and showcases the application of the approach to a sample use case: visualizing the sentiment the People's Daily conveyed toward Japan over time. This chapter also introduces a tool based on these methods that enables users to create visualizations of sentiment trends for any entity, even without knowledge of Chinese. Chapter 4 covers the results of the RAG experiment and highlights points of failure in the different RAG pipelines. Additionally, it presents the results of an analysis of language bias in an LLM-based evaluation. Finally, Chapter 5 discusses the limitations of the approaches taken to evaluation and suggests directions for future research.

doi:10.6342/NTU202502354

2 Literature Review

This literature review explores relevant research across three key areas. Section 2.1 reviews previous work on the People's Daily dataset, focusing on methods used prior to the widespread adoption of LLMs. Sections 2.2 and 2.3 discuss LLM-based methods for sentiment analysis and RAG, respectively. The insights from these sections lay the foundation for the experiments presented in this study.

2.1 Previous Studies on the People's Daily

The People's Daily, as the official newspaper of the Communist Party of China, has been the subject of numerous studies to understand its content, structure, and implications for political and social messaging.

Earlier studies were primarily qualitative analyses. For instance, Qian (1987) compared the structure and content of the People's Daily with the China Daily, a national English-language publication. This study described the typical structure of the People's Daily as follows: Page 1 typically featured transcripts of political speeches, editorials, and significant news events. Pages 2 to 4 focused on domestic news. The content of page 5 varied, consisting of, among others, philosophical and historical discussions and letters to the editor, while pages 6 and 7 covered international news. Page 8 was devoted to ar-

doi:10.6342/NTU202502354

ticles on culture and literature. The content was mainly authored by in-house reporters or sourced from the Xinhua News Agency. Overall, the author found the People's Daily focused mostly on domestic news.

Fang (2001) conducted a comparative study analyzing the coverage of protests in South Africa (1985) and Argentina (1989). By comparing articles from the People's Daily and the Kuomintang-owned Central Daily News, the author examined the choice of words, headlines, and recurring themes. The study found a stark contrast between the narratives during the South African protests, with the People's Daily using neutral language to describe the protesters' actions. At the same time, the Central Daily News referred to the events as "riots." However, both newspapers used similar terminology for the protests in Argentina, such as "riots" and "looting," indicating that the narratives were identical in this case.

As computational tools became more accessible, researchers began employing corpus-based methods and statistical analyses. Qian (2010) used keyword frequency analysis to study the discourse surrounding terrorism in the People's Daily and *The Sun*, a UK publication, before and after the September 11 attacks on the Word Trade Center in New York City. The study examined the frequency of different keywords and collocations of the words "terrorism," "terrorist," and "terror" in the two corpora.

In recent years, studies of the People's Daily have increasingly used machine learning techniques. Montiel et al. (2014) analyzed articles related to the Scarborough Shoal/ Huangyan Island territorial dispute, comparing coverage in the English-language edition of the People's Daily and the *Philippine Daily Inquirer*. This study tested the efficacy of machine learning algorithms (including Support Vector Machines, Naïve Bayes, and

doi:10.6342/NTU202502354

k-Nearest Neighbor) in classifying articles by source. The results showed high classification accuracy, with Naïve Bayes classifying all articles correctly. Principal Component Analysis revealed thematic differences between the two newspapers, reflecting divergent narratives about the territorial dispute.

Li and Hovy (2014) used a semi-supervised bootstrapping algorithm to perform entity-level sentiment analysis on the People's Daily for 60 years. Based on the assumption that sentiment toward an entity is somewhat consistent over time, the model iteratively expands the lexicon and sentiment predictions, starting with a small seed set of labeled sentiment words. The approach achieved a high correlation with historical diplomatic relations data.

Chan and Zhong (2019) introduced the "Policy Change Index" (PCI), a machine learning-based approach to predict significant policy changes based on the unexpectedness of the front page content of the People's Daily. According to their idea, a spike in the PCI would indicate an upcoming significant policy change. Comparing against major historical events, they found that the PCI did indeed indicate several big changes and did not make major false predictions. Still, not all major events were picked up on by the algorithm, which they ascribe to some changes coming without being discussed in the People's Daily beforehand, such as a major stimulus package in 2008.

2.2 Sentiment Analysis with LLMs

In recent years, LLMs have become a new tool for conducting sentiment analysis. While general sentiment analysis provides an overall assessment of the tone of a text, Rønningstad et al. (2022) showed that the sentiment conveyed by a text as a whole may not always align with the sentiment directed at individual entities, such as people or or-

doi:10.6342/NTU202502354

ganizations, that are mentioned in the text. LLMs, pre-trained on real-world data, bring both advantages and challenges to this task. Their ability to resolve entity coreferences (Chowdhery et al., 2022) may be beneficial for identifying overall sentiments toward entities in a longer document. However, Rønningstad et al. (2024) note that biases from generally positive or negative public opinion toward an entity in the training data can lead to skewed sentiment predictions. This section covers studies that have employed LLMs to perform sentiment analysis, focusing on those that identify sentiment toward individual entities. Terminology for such tasks varies widely across studies. Table 2.1 provides an overview of the task names used by the studies mentioned below.

Abbreviation	Full Task Name	Description	Reference
ABSA	Aspect-Based Sentiment Analysis	An umbrella term for ABSC and E2E-ABSA.	Wang et al. (2024)
ABSC	Aspect-Based Sentiment Classification	The LLM is tasked with identifying the sentiment toward a given aspect (which may or may not be an entity) mentioned in a sentence.	Wang et al. (2024)
E2E-ABSA	End-to-End Aspect-Based Sentiment Analysis	The LLM is tasked with identifying an aspect term and assigning a sentiment label in the same prompt.	Wang et al. (2024)
ELSA	Entity-Level Sentiment Analysis	The task of identifying sentiment toward a specific entity at the document level.	Rønningstad et al. (2024)
MEBSA	Multimodal Entity-Based Sentiment Analysis	An umbrella term for tasks related to finding entities and classifying sentiment toward them in image/ text combinations.	Yang et al. (2024)
MESC	Multimodal Entity-based Sentiment Classification	The specific task of identifying sentiment toward a given entity in a multimodal context (image/text).	Yang et al. (2024)
MESPE	Multimodal Entity-Sentiment Pair Extraction	The LLM is tasked with extracting entities and identifying sentiment toward them, similar to E2E-ABSA but in a multimodal context.	Yang et al. (2024)

Table 2.1: Overview of terminology used for tasks related to identifying sentiment toward an entity. This list is not exhaustive and only includes names used in studies discussed in this section.

doi:10.6342/NTU202502354

Wang et al. (2024) compared ChatGPT to fine-tuned BERT and state-of-the-art supervised models across various sentiment analysis tasks, including ABSC and E2E-ABSA, which differ only in whether or not the target entities are given in the prompt. They found that ChatGPT performed better than BERT but lagged behind state-of-the-art models, with few-shot prompting leading to improved results for aspect-based sentiment classification.

Yang et al. (2024) proposed an approach for multimodal sentiment analysis on data combining text and images. The method consists of two key steps. First, an image caption is generated, along with a list of entities depicted and a label for the overall sentiment conveyed by the image. These elements are then summarized into a single sentence, which is used in the final prompt instead of the image itself. In the second step, similar image/text combinations are retrieved from a database and used as few-shot examples. Depending on the task, the final prompt either asks the LLM to generate the overall sentiment for a given entity (MESC) or to extract entities along with their sentiment labels (MESPE).

2.2.1 Multilingual Studies

For entity-level sentiment analysis at the document level, Rønningstad et al. (2024) tested `gpt-4` on a Norwegian dataset using a five-point sentiment scale. They explored various prompting strategies, including Norwegian versus English prompts and few-shot examples, and compared the results to those of human annotators. Their findings highlight the subjectivity of sentiment analysis, with low inter-annotator agreement among human annotators. The best results were achieved using prompts tailored to the language of the dataset (Norwegian), combined with few-shot examples and more detailed instructions .

doi:10.6342/NTU202502354

Buscemi and Proverbio (2024) evaluated multiple LLMs, including `gpt-4`, `gpt-3.5-turbo`, `gemini-1.0-pro`, and `llama-2-7b`, on their ability to assess the sentiment of sentences in ten languages. The task required the models to rate how the author of a sentence might have felt on a scale from 1 to 10. `llama-2-7b` consistently gave ratings around 8, regardless of input, while the other models produced results that more closely matched human judgments. Differences between languages were observed, such as `gpt-4` assigning higher scores to Chinese sentences compared to other languages. While not focused on sentiment analysis at the entity level, this study is noteworthy for being among the few that evaluated multilingual data.

2.3 Retrieval-Augmented Generation

Retrieval-Augmented Generation (RAG) combines two key steps: *Retrieval* refers to fetching relevant information from external sources, such as a database or document collection. During the *generation* step, the retrieved information is used as context in a prompt to help an LLM to produce a relevant response.

The term Retrieval-Augmented Generation was first introduced by Lewis et al. (2020). To address the typical problems encountered when using LLMs for question answering tasks (hallucinations, lack of source annotations and difficulty updating stored information), they built a system where chunks of texts retrieved from Wikipedia are added to the input sequence fed into BART, an early-generation LLM (Lewis et al., 2019). This approach outperformed both closed-book models and other retrieval-based approaches on a variety of tasks including open-domain question answering, fact verification and question generation.

doi:10.6342/NTU202502354

Since its introduction, RAG has been adapted for a variety of use cases, with numerous papers proposing modified architectures tailored to specific applications (Gao et al., 2024). This section covers a small selection of architectures, use cases, and evaluation methods found in recent literature.

2.3.1 Architectures Using Knowledge Graphs

Chaudhri et al. (2022) define a knowledge graph as a "directed labeled graph in which domain-specific meanings are associated with nodes and edges."

One way in which knowledge graphs have been incorporated into RAG architectures is as a datastore. In this architecture, instead of embedding the user query and finding similar text chunks in a vector store, the retrieval step involves an LLM generating a statement in a query language to retrieve relevant nodes, edges, paths, and other information from a graph database. In the generation step, this information is then used as context in a second prompt to enable the LLM to give a relevant answer.

This approach has been explored in several studies. One of the first to investigate related tasks, Guo et al. (2023), evaluated `gpt-3.5-turbo-instruct` on various graph-related reasoning tasks, including query generation and question answering. They found performance highly dependent on the prompting strategy, with some setups outperforming state-of-the-art approaches.

An example of this type of architecture applied to a task is Kulkarni et al. (2024), who applied a knowledge graph-based RAG architecture to a clinical trial dataset, achieving promising results with `GPT-4`. The graph database contains nodes such as 'Study,''Disease,' and 'Symptom,' with connections such as 'investigated_in' linking diseases to studies, and ‘presents’ connecting diseases to symptoms. In their implementation, they used the *LangChain* `GraphCypherQAChain` (Chase, 2022), which employs a zero-shot prompt to generate a Cypher query statement based on the user query and graph schema during the retrieval step, while a one-shot prompt is used for answer generation.

Another way knowledge graphs have been used in RAG systems is as a tool for identifying connections within text data during the indexing process.

Notably, Edge et al. (2024) introduced “GraphRAG,” an approach where a datastore is created from the text by extracting entities, relationships, and claims using an LLM, followed by community detection on the resulting knowledge graph. Specifically, they use the Leiden algorithm (Traag et al., 2019) to iteratively group highly interconnected nodes into clusters that may correspond to a common topic or theme. Finally, the LLM then writes summaries of different levels of communities. For retrieval, the user chooses which community level to use; higher-level summaries tend to provide more comprehensive answers, while lower-level communities often offer more concrete details. All community summaries at the user-specified level are retrieved for any question. In the generation step, the community summaries are shuffled and split into smaller chunks. The LLM generates answers for each chunk, scores each answer for helpfulness, and combines the top-scoring answers into a final response. In a pairwise evaluation for different criteria, the authors found that using higher-level communities for answer generation yields more comprehensive and diverse answers, while the answers given by a classic vectorstore-based system scored higher on directness. When evaluating for *empowerment*, that is, how useful the answer might be to a user, results were mixed.

doi:10.6342/NTU202502354

2.3.2 Multilingual Applications

While most of the literature on RAG has focused on English-only applications across the datastore, retrieval, and prompts, there has been some, albeit limited, research on multilingual applications. Chirkova et al. (2024) conducted a comprehensive study on the impact of language variation and mixing at different stages of the RAG pipeline. They examined the effects of using English, the query language, or multilingual data in the datastore; English or the query language in system prompts, with or without explicit specification of the output language; and English-based vs. multilingual models for the retriever and LLM. Overall, their findings indicated that a multilingual retriever effectively handles a foreign-language datastore, eliminating the need to translate user queries before performing retrieval. Additionally, translating the system prompt into the language of the query and explicitly specifying the output language while using a multilingual LLM improved performance considerably, particularly in ensuring the output matched the language of the query.

2.3.3 Evaluation Methods and Criteria

Evaluation of RAG architectures is not a straightforward task as what makes a "good" RAG system is highly dependent on the objectives of the individual use case. Evaluation methods are therefore usually tailored to the specific purpose of the system. Even in the small selection of studies referenced above, there is great variation between the approaches to evaluation (see Table 2.2).

doi:10.6342/NTU202502354

Method	Task Type	Reference
Manual	Jeopardy Question Generation	Lewis et al. (2020): Pairwise comparison of output against BART without RAG by Factuality and Specificity
Comparison against reference answer	Open-Domain QA	Lewis et al. (2020): Exact Match scores
		Chirkova et al. (2024): Trigram Recall
	Abstractive QA	Lewis et al. (2020): BLEU and ROUGE-L
	Fact Verification	Lewis et al. (2020): Label accuracy on FEVER (Thorne et al., 2018)
	Jeopardy Question Generation	Lewis et al. (2020): Q-BLEU (Nema and Khapra, 2018)
LLM as a judge	Global Sensemaking QA	Edge et al. (2024): Pairwise comparison of answers by four criteria: Comprehensiveness, Diversity, Empowerment, Directness
	Domain-specific QA	Kulkarni et al. (2024): RAGAs Framework (Es et al., 2023)

Table 2.2: Summary of evaluation methods used in studies cited in this literature review.

Manual evaluation was conducted only by Lewis et al. (2020) for the *Jeopardy Question Generation* task, which tests the ability of a system to generate a question to a given answer. For this task, the output quality depends on two criteria: Factuality of the QA pair and specificity of the question, i.e., whether the answer is uniquely correct for the generated question. Both of these criteria assess the performance of the system as a whole, without separately evaluating the retrieval and generation components. Annotators were tasked with comparing two answers at a time to determine which was more factual or specific, with 'both' or 'none' as valid options. They were encouraged to consult external online sources to verify their judgments.

A more common approach is to evaluate output against a reference answer. With the exception of the Fact Verification task used by Lewis et al. (2020), which required the LLM to choose from pre-defined labels and could therefore simply be compared against a gold

doi:10.6342/NTU202502354

standard, all such tasks were evaluated based on various types of string matching metrics. In the only multilingual study in the lineup, Chirkova et al. (2024) employed character-level trigram recall, which measures the overlap of consecutive character triplets between the correct answer and the generated response. This method ensures that minor spelling differences in entity names across languages are penalized proportionally, rather than being considered entirely incorrect. However, they note that this approach has limitations when working with languages that do not use the Latin alphabet.

LLM-based approaches were employed in two studies. Edge et al. (2024) performed a pairwise evaluation of answers generated by different setups of GraphRAG to the same question, asking the LLM to select the better answer based on four specified criteria. The only study evaluating retrieval and generation separately was Kulkarni et al. (2024), which used the RAGAs framework. This framework evaluates three key metrics: *Context Relevance*, which assesses the retrieval step and penalizes irrelevant details in the retrieved context; *Faithfulness*, which assesses the generation step and determines to what extent the answer is correct given the retrieved context; and *Answer Relevance*, which evaluates overall output quality by measuring the cosine similarity between the embeddings of the question and the answer, a method that does not rely on an LLM. As noted by Chirkova et al. (2024), there is a lack of research on LLM-based evaluation methods in multilingual contexts.

doi:10.6342/NTU202502354

doi:10.6342/NTU202502354

3 Sentiment Analysis Experiment

This chapter focuses on experiments conducted to find the optimal setup for a tool that visualizes sentiment conveyed toward entities over time in the People's Daily.

Li and Hovy (2014), who pursued a similar goal, noted that sentiment analysis on the People's Daily is particularly challenging due to the frequent use of metaphors, proverbs, and nicknames. Using a bootstrapping approach, they were able to infer the sentiment of such expressions well enough to generate reliable sentiment data for entities over time. This chapter explores whether a simpler, prompt-based method can achieve comparable results, eliminating the need for intermediate steps such as extracting sentiment-carrying expressions and excluding sentences with multiple sentiment targets.

The approach taken in this study closely follows that of Rønningstad et al. (2024), who addressed the ELSA task with goals similar to those of this study and demonstrated promising results. Experiments were conducted to identify the optimal LLM setup, prompt language, and configuration for sentiment analysis. In addition, a qualitative evaluation was performed by computing sentiment scores for a dataset of articles about Japan, assessing whether sentiment spikes corresponded to significant events in Sino-Japanese history and whether these patterns could be captured in a time series model.

doi:10.6342/NTU202502354

The resulting tool, built based on these experimental findings, allows users to input an entity name and a time frame to generate a sentiment-over-time plot using document-level sentiment scores.

3.1 Test Set Creation

For the test dataset, ten articles were randomly selected from the corpus, with the constraint that each article must not exceed 1000 characters. The relatively small number of articles and the character limit were chosen to minimize participant workload, since each participant had to read every article.

Entities were extracted from the texts using SpaCy, which I chose due to its ease of use, its reasonable performance in initial tests, and its categorization of entities into pre-defined types, which was useful for further filtering. The entities were filtered to the following types:

- PERSON: Individuals
- ORG: Organizations
- GPE: Geopolitical entities (such as countries and cities)
- NORP: Nationalities, religious and political groups

Some entities were removed from the list of entities to be considered when a text contained a large number of extracted entities in order to reduce participant workload.

doi:10.6342/NTU202502354

3.2 Human Annotation

3.2.1 Questionnaire

The selected texts were inserted into a Google Form, preserving their original Simplified Chinese. Each text was accompanied by a set of 1 to 10 questions ("What is the overall sentiment toward <entity>?") and corresponding answer options (the same as those used by Rønningstad et al., 2024):

- Positive-Standard
- Positive-Slight
- Neutral
- Negative-Slight
- Negative-Standard

A total of 47 questions were included, with questions and answer options provided in English, Traditional Chinese, and Simplified Chinese. Full participant instructions are included in Appendix A.1.

3.2.2 Participants

The questionnaire was completed by 9 participants, all of whom were native Chinese speakers and proficient in reading Simplified Chinese characters. However, all participants reported being more familiar with Traditional characters than Simplified ones. The participants' ages ranged from 23 to 33 years, with an average age of 25.9 years. Five participants had a background in Linguistics, while four did not.

Participant	In Majority	Within Range	Out of Range
Participant 1	0.702	0.213	0.085
Participant 2	0.511	0.298	0.191
Participant 3	0.851	0.128	0.021
Participant 4	0.596	0.362	0.043
Participant 5	0.851	0.149	0.000
Participant 6	0.660	0.255	0.085
Participant 7	0.745	0.234	0.021
Participant 8	0.766	0.213	0.021
Participant 9	0.872	0.128	0.000

Table 3.1: Fractions of responses for each participant: aligned with the majority, within the range of others' answers, or identified as outliers.

3.2.3 Inter-Annotator Agreement

Table 3.1 shows three different metrics[1]: "In Majority" specifies the proportion of participant responses that matched the *majority answer* (i.e., the answer given by the highest number of participants overall). Since this metric does not take into account the distribution of the remaining answers, "Within Range" and "Out of Range" specify the proportion of responses that fell within or outside of the range of answers provided by other participants, respectively. An example of the "Within Range" metric would be if a participant answered "Negative-Slight" and all other participants answered "Negative-Standard" or "Neutral." Although this answer would be in the minority, it would not be considered an outlier because it falls within the range of the other responses. The results show that while there was substantial disagreement between participants, with "In Majority" rates between 55.1% and 87.2%, only one participant gave a particularly high number of outlier answers (Participant 2 with 19.1%). In comparison, the next-highest rate is 8.5%.

[1]The only difference between my approach and that of Rønningstad et al. (2024) is in the definition of "In Majority." While they considered an answer to align with the majority if it matched the majority answer(s) given by other participants, I defined the majority answer as the response most commonly given by all participants, including the participant themselves.

doi:10.6342/NTU202502354

Figure 3.1 shows the variation in participant responses for each entity as box plots. Only one entity has an interquartile range that crosses the ‘Neutral’ line, indicating that while annotators may disagree on the strength of the sentiment, they tend to agree on its overall positivity or negativity.

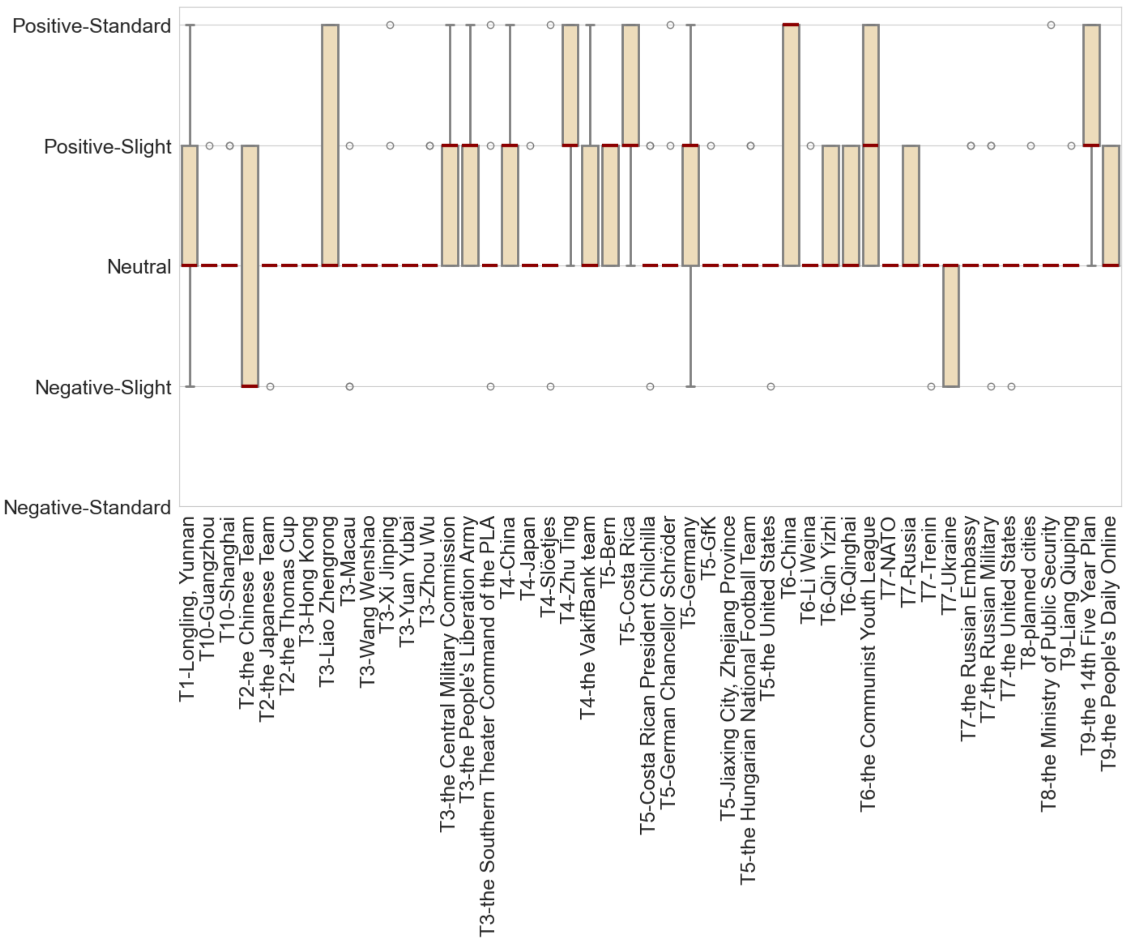


Figure 3.1: Boxplots showing the distribution of entity labels assigned by the human annotators for each entity.

doi:10.6342/NTU202502354

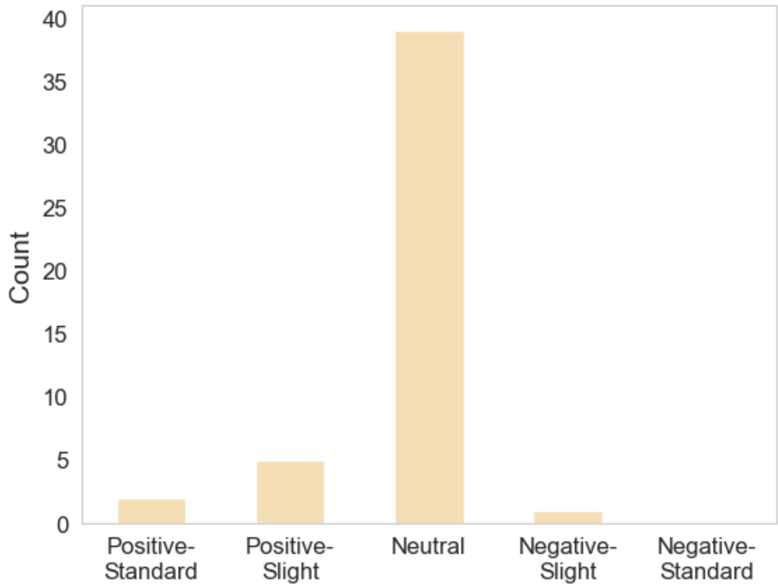


Figure 3.2: Number of occurrences of each sentiment label in the gold standard dataset.

3.3 Manual Curation of a Gold Standard

Following Rønningstad et al. (2024), the responses were manually curated to create a gold standard dataset using the following criteria:

1. The label chosen by the majority of participants becomes the gold standard.
2. In case of a tie, the minority responses are used as a tiebreaker to identify an overall tendency.

   (Hypothetical example: 1 'Positive-Standard', 4 'Positive-Slight', 4 'Neutral' → 'Positive-Slight' becomes the gold standard).
3. Where the remaining responses cannot be used to find a tendency, the one closer to the 'Neutral' label becomes the gold standard.

doi:10.6342/NTU202502354

Step 2 only became necessary in one case, while Step 3 never became necessary. As shown in Figure 3.2, the gold standard dataset[2] had 'Neutral' as the most common label for the majority of entities.

3.4 Annotation With LLMs

3.4.1 Implementation

A total of 24 setups were implemented, with the following variables: LLM (`gpt-4o` vs. `gemini-1.5-flash`), prompt format (single entity per prompt vs. batch prompt, i.e., multiple entities per prompt), prompt language (German, English, vs. Chinese), and whether or not additional guidance was provided to the LLM. For German and English prompts, additional guidance refers to an additional sentence being added to each prompt that informs the LLM that the entity name will be given in English/German while the text will be in Chinese. For Chinese prompts, additional guidance means that the terms *sentiment analysis* and *entity* appeared in English in the system instructions.

I chose not to include LLaMA2 in the LLM lineup, as Buscemi and Proverbio (2024) demonstrated its poor performance on sentiment analysis (see Section 2.2).

The English system instructions and single entity prompt are given below. The remaining prompts can be found in Appendix A.2.

[2] The full dataset including the relevant texts and all participant responses is available at `https://docs.google.com/spreadsheets/d/1GPDOAZZIvq0JFo9mqIqe1LOx8bglSONjJDPnrVRhTUg/edit?usp=sharing`. Due to a technical error in Google Forms, some questions had been duplicated in the questionnaire. While participants usually repeated their answer for the second instance of the same question, there were two instances where a participant gave a different answer (off by one category) the second time. Overall, however, there were no differences in the outcomes for the gold standard labels. In all cases, only the first set of responses was included in the final dataset.

doi:10.6342/NTU202502354

Prompt 3.1: English System Instructions for Sentiment Analysis (Single Entity)

```
You are a helpful assistant designed to output sentiment
classification labels. All questions are about entity-wise
sentiment analysis on Chinese texts. You will analyze the
sentiment toward the given volitional entity, based on a Chinese
text that will be provided to you in a prompt. The reply should
be the assigned label, one of `['Positive-Standard', 'Positive-
Slight', 'Neutral', 'Negative-Slight', 'Negative-Standard' ]`.
'Neutral' is the label assigned when you cannot identify any
sentiment toward the entity in question. 'Positive-Slight' and
'Negative-Slight' are used if an entity receives slight, vague
or uncertain sentiment. Otherwise, the 'Positive-Standard' and
'Negative-Standard' labels are used for all clear sentiments
expressed towards the entity. You should not refer to common
knowledge about an entity, but strictly analyze the sentiment
conveyed in the given text. If both positive or negative
sentiments exist, you must decide what is the prevalent or
overall strongest sentiment conveyed in the text regarding the
entity in question.
The output should be a JSON formatted formatted in the following
schema:
{
"label": string // The label assigned to the entity in
question, one of ['Positive-Standard', 'Positive-Slight',
'Neutral', 'Negative-Slight', 'Negative-Standard' ]. If you
could not find the entity in the text, write 'none'.
}
```

Prompt 3.2: English Prompt Template for Sentiment Analysis (Single Entity)

```
We are going to analyze the following text: "{text}"
Your task is to assign a sentiment label that the text
communicates regarding "{entity}", according to the system
instructions for the assistant.
# In the "Additional Guidance" condition
+ Note that the entity names are given in English, while they
will likely appear in Chinese in the text.
```


doi:10.6342/NTU202502354

3.4.2 Results

In total, 1128 API calls were made for the single-entity prompts, and 240 API calls for the batch prompts (a single prompt per text, asking about all entities at once).

Single-entity prompts failed to assign a sentiment label in 18 cases, while batch prompts returned the ‘Unknown’ label 11 times. Additionally, in six instances, `gemini-1.5-flash` failed to assign a label for an entity within a batch prompt, highlighting the issue of entities potentially being overlooked when multiple entities are included in the same prompt. Instances of no label being assigned were spread across various setups, with one setup (Gemini - Chinese Prompts - Batch Prompts - No Additional Guidance) standing out, as 10 entities were left unlabeled.

Table 3.2 presents the same metrics for the LLMs as those calculated for human annotators (see Table 3.1), but with the performance of the LLMs evaluated against human annotators rather than one another. While performance varies, it is immediately clear that "Out of Range" answers are much more common, with no setup falling below the 10% mark. Notably, one setup (Gemini - Chinese Prompts - Batch Prompts - Additional Guidance) reached an outlier rate of 31.9%.

To evaluate the different setups against the gold standard, I calculated the Weighted Cohen's Kappa with quadratic weights for each system compared to the curated labels. While Rønningstad et al. (2024) used Cohen's Kappa, I chose the weighted version to ensure that the distance from the gold standard was accounted for. The results are displayed in Figure 3.3.

doi:10.6342/NTU202502354

Key observations include:

- The highest overall agreement with the gold standard was achieved with Gemini, using Chinese prompts annotated with additional English keywords and asking participants to rate a single entity at a time.

- Gemini particularly struggled with batch prompting; all batch prompting setups performed worse than the lowest-performing setup for single entity prompts.

- It is difficult to draw a clear conclusion regarding prompt language. For Gemini, Chinese worked best for single entity prompting but worst for batch prompting. For batch prompting, German performed best, with English close behind. For GPT-4, the best setup used English for both single entity and batch prompts.

- The impact of additional guidance was unclear. It seemed to help in some cases, hinder in others, and had little effect in some setups.

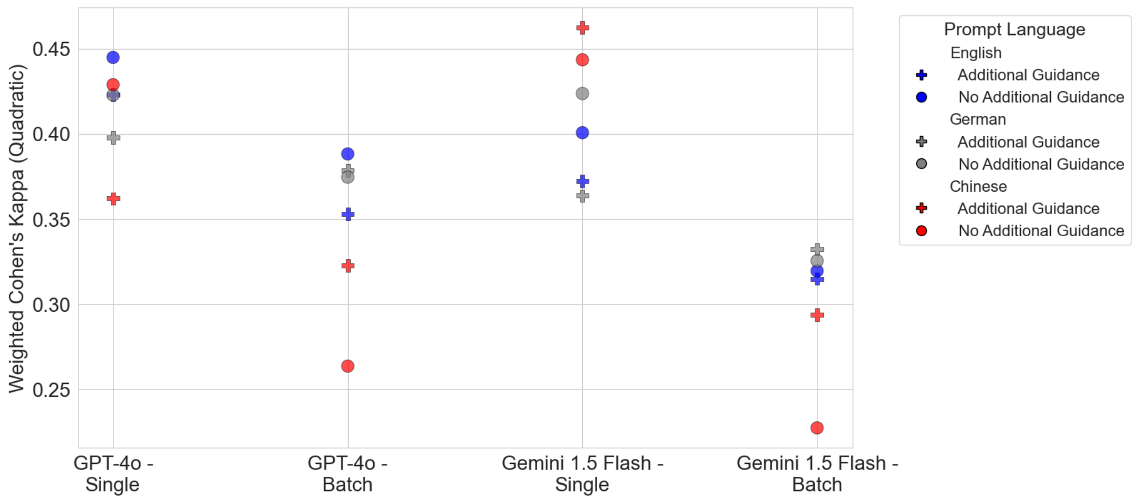


Figure 3.3: Weighted Cohen's Kappa (Quadratic) by system, compared against the gold standard labels.

doi:10.6342/NTU202502354

LLM	# of Entities	Prompt Language	Additional Guidance	In Majority	Within Range	Out of Range
gpt-4o	Single	English	×	0.638	0.213	0.149
gemini	Single	English	×	0.596	0.191	0.213
gpt-4o	Single	English	✓	0.660	0.191	0.149
gemini	Single	English	✓	0.574	0.191	0.234
gpt-4o	Batch	English	×	0.638	0.213	0.149
gemini	Batch	English	×	0.468	0.362	0.170
gpt-4o	Batch	English	✓	0.596	0.234	0.170
gemini	Batch	English	✓	0.468	0.383	0.149
gpt-4o	Single	German	×	0.553	0.149	0.298
gemini	Single	German	×	0.574	0.170	0.255
gpt-4o	Single	German	✓	0.596	0.170	0.234
gemini	Single	German	✓	0.489	0.191	0.319
gpt-4o	Batch	German	×	0.617	0.255	0.128
gemini	Batch	German	×	0.426	0.319	0.255
gpt-4o	Batch	German	✓	0.617	0.234	0.149
gemini	Batch	German	✓	0.404	0.362	0.234
gpt-4o	Single	Chinese	×	0.532	0.319	0.149
gemini	Single	Chinese	×	0.638	0.191	0.170
gpt-4o	Single	Chinese	✓	0.596	0.255	0.149
gemini	Single	Chinese	✓	0.660	0.213	0.128
gpt-4o	Batch	Chinese	×	0.681	0.255	0.064
gemini	Batch	Chinese	×	0.468	0.277	0.255
gpt-4o	Batch	Chinese	✓	0.681	0.255	0.064
gemini	Batch	Chinese	✓	0.468	0.319	0.213

Table 3.2: Performance metrics by LLM, number of entities per prompts, prompt language, and whether or not the LLM received additional guidance. The `gemini` system used was `gemini-flash-1.5`.

3.4.3 Application to a Large Dataset

Since assigning labels to a single text is an inherently subjective task, as evidenced by the low inter-annotator agreement among human annotators, this approach was further evaluated at a macro scale to determine whether it could reveal overall trends in a large dataset. While individual article sentiment may vary or be contentious, the aim was to assess whether consistent patterns would emerge when analyzing a substantial number of texts. To this end, the method was applied to texts mentioning the same entity (Japan) to identify expected trends. Sino-Japanese relations are well-documented, and as a government-owned publication, more negative coverage would be anticipated during periods of heightened tensions, with more positive articles during times of improved relations, such as when treaties were signed between the two countries.

To carry out this evaluation, all articles mentioning Japan were sampled by performing a keyword search for “日本.” Using the best-performing setup from the previous evaluation (Gemini - Chinese Prompts - Single Entity Prompts - Additional Guidance), 97.5% of the 133,435 articles were successfully assigned a sentiment label.

Figure 3.4 visualizes the distribution of sentiment over time. As anticipated, clusters of negative articles appear in the period after World War II and, starting in the 1980s, around the ten-year anniversaries of the war’s end. The initial trend of predominantly negative reporting diminishes around 1972, coinciding with the normalization of diplomatic relations between China and Japan.

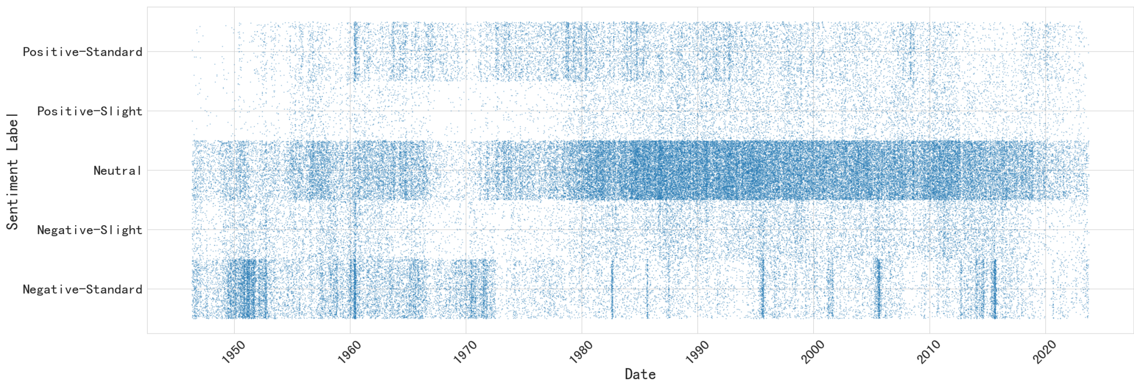


Figure 3.4: Scatterplot of the sentiment toward Japan in 130086 articles, with jitter applied to minimize overlapping. Each point represents the sentiment conveyed toward Japan in a single article. Positive coverage is rare before the normalization of the relationship between China and Japan in the 1970s. Since the 1980s, negative coverage appears to have been especially common around the anniversaries of the end of World War II. A bar chart of these values aggregated by year is included in Appendix A.4.

doi:10.6342/NTU202502354

While some patterns, such as the WWII round anniversaries in the Japan dataset, are visible in a scatterplot, subtler effects can be harder to detect. To explore these in more detail, I employed *Facebook Prophet* (Taylor and Letham, 2018), a time series model. Time series models are designed to analyze data ordered over time, making them particularly useful for identifying trends and detecting deviations from expected patterns that might not be immediately visible in raw plots. Unlike scatterplots, which provide a static snapshot of variation, time series models incorporate the sequence and spacing of data points to uncover underlying structures such as recurring shifts, gradual changes, or irregular spikes. This makes them well-suited for revealing subtle effects that may be hidden in visually cluttered or noisy datasets. I chose Facebook Prophet specifically for its intuitive built-in visualization tools.

Since time series models require numerical data, I mapped the sentiment labels to numerical values as shown in Table 3.3 and calculated a daily sentiment score by summing all the sentiment values for a given day.

Sentiment Label	**Numerical Value**
Positive-Standard	2
Positive-Slight	1
Neutral	0
Negative-Slight	-1
Negative-Standard	-2

Table 3.3: Mapping of sentiment labels to numerical values.

The average daily sentiment score was -0.891, with a standard deviation of 3.882, indicating a slight overall negativity but significant daily variation.

doi:10.6342/NTU202502354

A *Prophet* model was trained on this data using default parameters, incorporating four types of "holidays", i.e., special events where deviation from the usual pattern would be expected:

- Round Anniversaries of the end of World War II (every five years on August 15, from 1945)
- Other anniversaries of the end of Word War II: (August 15, all other years)
- Signing of the Japan−China Joint Communiqué [3] (September 29, 1972)
- Anniversaries of the signing of the Japan-China Joint Communiqué: (every year on September 29, from 1973)

Cross-validation with a 730-day initial training period, 180-day periods, and a 365-day horizon produced an RMSE of 3.79, comparable to the standard deviation of the dataset, suggesting reasonable but not perfect predictions. The model achieved a Coverage of 0.84, meaning that 84% of actual values fell within its prediction intervals.

Figure 3.5 displays the daily sentiment scores alongside the trendline generated by the *Prophet* model, incorporating the specified holidays. As anticipated, anniversaries of the end of World War II show spikes in negative reporting, with round anniversaries producing larger spikes. Similarly, the normalization of relations in 1972 corresponds to a positive spike, followed by smaller positive spikes on subsequent anniversaries of the event.

Figure 3.6 isolates trends by the overall trend, holiday effects, weekly trends and changes over the course of the year (we can see the effects of the aforementioned holidays

[3] The *Joint Communique of the Government of Japan and the Government of the People's Republic of China* marks the normalization of the diplomatic relationship between the two countries.

doi:10.6342/NTU202502354

even here, with mid-August being the most negative and late September the most positive). Notably, the overall trendline mirrors the two significant shifts seen in the Japan trendline generated by Li and Hovy (2014): an improvement from overall negative sentiment to neutral or somewhat positive sentiment in the early 1970s, followed by a smaller decline in the mid-1990s.

The uniform appearance of the spikes in the trend line arises from how *Prophet* models holiday effects. *Prophet* treats holidays as recurring events with consistent impacts on the time series, estimating their effect based on the data and applying it uniformly across similar events in the future. Variability between individual occurrences of an event is not considered.

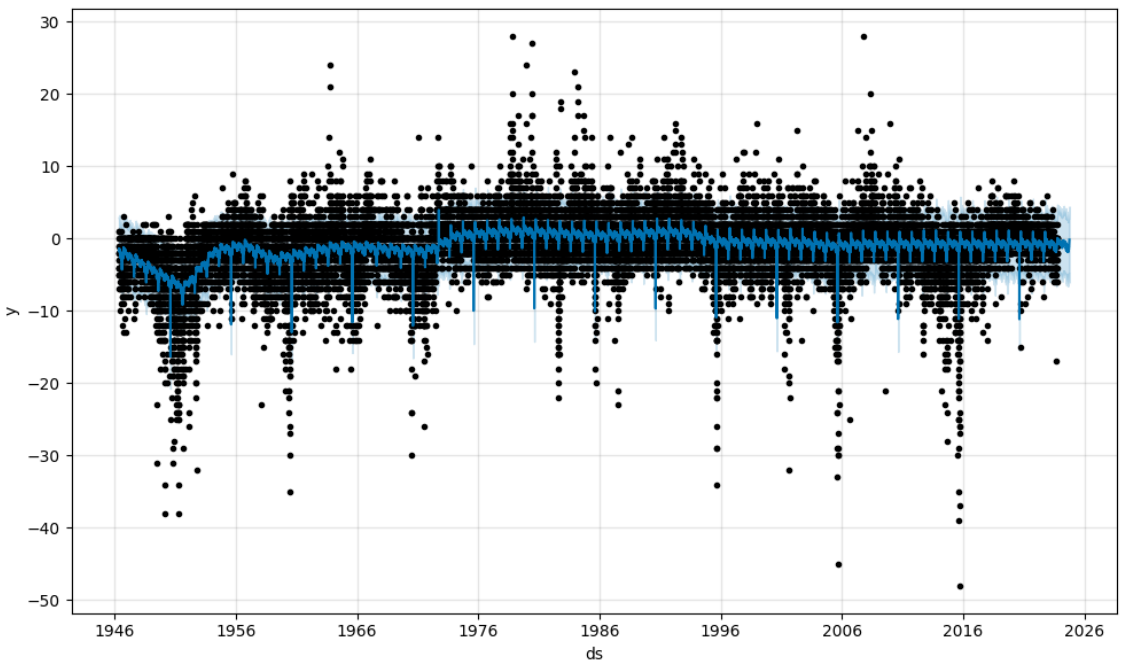


Figure 3.5: Time series analysis by *Facebook Prophet* on the Japan dataset: The scatterplot represents the sums of sentiment scores on each day, while the blue trendline combines overall, monthly and weekly trends as well as holiday effects (in this case, the anniversaries of the end of WWII and the signing of the Japan-China Joint Communique).

doi:10.6342/NTU202502354

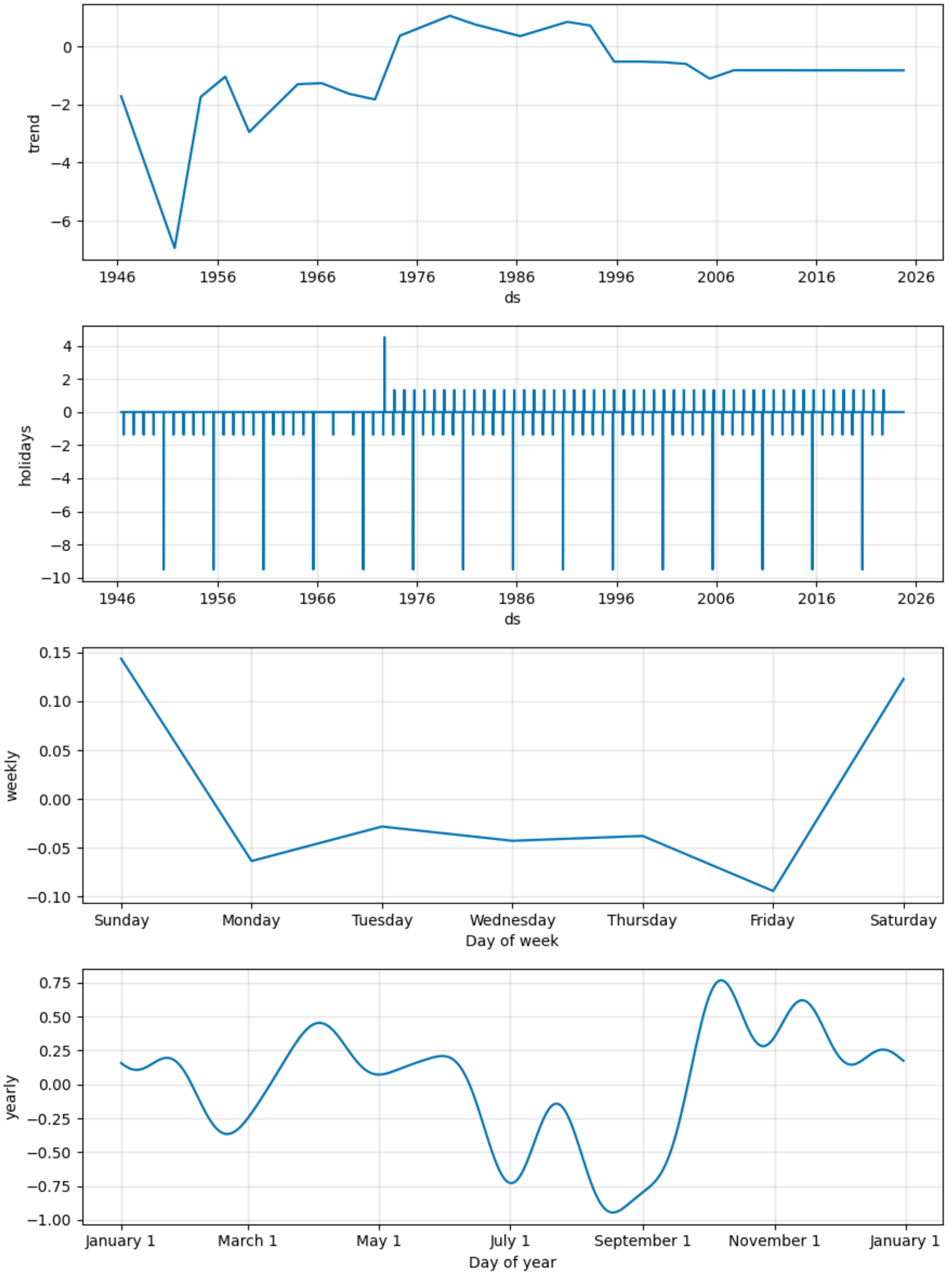


Figure 3.6: Time series analysis by *Facebook Prophet* on the Japan dataset: Overall trend, holiday effects, weekly and yearly trends. No holiday effects are observed on 1966-08-15 and 1968-08-15 because of a lack of data for those dates.

doi:10.6342/NTU202502354

3.5 Summary of Findings

1. Human annotators showed substantial disagreement about the exact labels (same finding as in Rønningstad et al. (2024)). However, they usually disagreed on the strength of the sentiment, not on its polarity.

2. In a comparison against human-annotated labels, including a single entity per prompt led to better results than asking the LLM to annotate all entities included in the text at once. This was especially true for `gemini-1.5-flash`.

3. The exact effects of the prompt language and the level of detail in the instructions remain unclear as there was no clear pattern in the results.

4. The overall best-performing setup was to prompt Gemini in Chinese, only asking about a single entity at a time and with the prompt including the English words *sentiment analysis* and *entity*.

5. Application of the method to a large dataset of articles mentioning Japan showed expected patterns in the data.

6. A *Facebook Prophet* model trained on the LLM-generated sentiment data for Japan produced a trendline with the same major patterns as the one generated by Li and Hovy (2014), indicating that the method is indeed suitable.

doi:10.6342/NTU202502354

3.6 Sentiment Analysis Tool

The final version of the tool[4] uses the setup that showed the best performance in the above experiment (prompting Gemini in Chinese, providing a single entity at a time and including the English words *entity* and *sentiment analysis*). The user can input an entity in any language. An API call is made to `gpt-4o-mini` to generate a list of possible Simplified Chinese translations for the entity name. These translations are then displayed to the user. If the user knows Chinese, they can remove irrelevant translations or add additional ones. The prompts used to find the translations and their synonyms are provided in Appendix A.3.

The final list of synonyms is used to perform a keyword search of the corpus for texts that contain at least one of the synonyms within a user-specified time frame. The user can then choose a number of articles to randomly sample from the corpus to create the prompt. These selected articles are shown in a table.

Next, the user selects the synonym to use in the search prompt. By default, this is the first (and most likely best) translation in the list to accommodate non-Chinese speakers.

Finally, a scatter plot showing sentiment over time based on the selected articles is generated, along with a LOESS smoothing line (see Figure 3.7 for an example).

[4] The final version of the tool is available at `https://github.com/deborahwatty/peoples_daily_entity_sentiment/`

doi:10.6342/NTU202502354

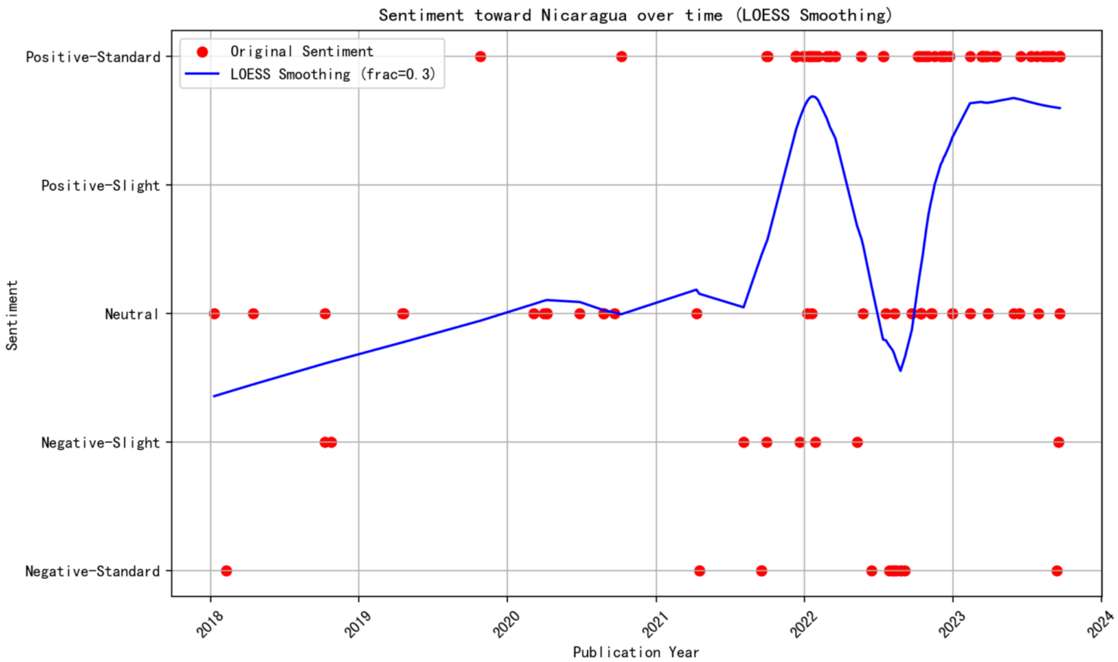


Figure 3.7: Sentiment in the People's Daily toward Nicaragua between 2018 and 2023, as generated by the final sentiment tool. Notably, a cluster of positive articles stands out starting in 2021, coinciding with the resumption of diplomatic relations between Nicaragua and China.

doi:10.6342/NTU202502354

4 Retrieval-Augmented Generation Experiment

While the tool introduced in the previous chapter can provide an overview of trends in the polarity of opinions expressed by a newspaper, it does not provide the user with any information on the content of the texts used to generate the sentiment data.

RAG has been employed successfully for finding and aggregating information in text data. However, the literature describes a wide range of architectures, many of which have only been tested on specific types of questions that the architecture was designed to answer, and very few having been applied to multilingual tasks. This chapter describes an experiment designed to evaluate the performance of three Retrieval-Augmented Generation (RAG) architectures and their multilingual capabilities:

- A classic vectorstore-based system (referred to as *VectorRAG* from this point on). Answers are generated based on three text chunks from the corpus that are similar to the user query. Due to its design, it is expected to perform best on questions that focus on specific details contained within a single article, hereafter referred to as *Detail questions*.
- A system that queries a graph database directly (hereafter called *CypherRAG*, re-

doi:10.6342/NTU202502354

flecting its use of Cypher for retrieval). This system is designed to find connections in a knowledge graph and generate answers based on retrieved nodes and relationships, which is why it is expected to perform best on questions which require identifying paths between entities in the dataset (hereafter referred to as *Connection questions*).

- GraphRAG with Global Search (retaining the name *GraphRAG* here for consistency with the original publication (Edge et al., 2024)). As introduced in Section 2.3.1, it generates responses based on community summaries generated from a knowledge graph during indexing. It is therefore expected to perform best at answering questions which ask for broader insights about the dataset as a whole (hereafter referred to as *Big Picture questions*.

The experiment is designed to address the following questions:

- How well does each system perform on the type of question for which it is designed vs. other types of questions?
- How does changing the language of the dataset and/or user queries affect system performance?
- What types of mistakes is each system likely to make?

The three question types (Detail, Connection, and Big Picture) were selected to align with the core design strengths of each system, allowing each to serve as a baseline for its respective category. In existing literature, RAG architectures are typically evaluated only on the specific type of question they were designed to answer, leaving open the question of how well they generalize to other query types (see Section 2.3). This experiment addresses that gap by assessing the performance of each system performance not only on its target

doi:10.6342/NTU202502354

question type but also on the others, thereby testing their potential as general-purpose QA tools.

In the following sections, I begin by explaining the selection of the data subset used for the experiments, followed by the implementation details of each architecture. Next, I describe the process of generating test questions and the criteria used to evaluate each question type. Detail and Connection questions are evaluated manually, while Big Picture questions are assessed automatically using an LLM.

The evaluation results are presented as follows: First, for each question type, I compare the performance of different architectures, demonstrating that each system performs best on the question type that was designed for it. This includes an analysis of the effects of changing the language of the database and/or query. Second, for each architecture, I provide an error analysis, identifying patterns such as whether mistakes occurred during retrieval or generation and common types of errors. Finally, I present the results of a supplementary evaluation examining whether language bias influences LLM-based evaluation methods in a multilingual context.

4.1 Dataset

The dataset for the RAG experiment is a subset of the People's Daily corpus, limited to the 4253 unique articles from January and February 2020 due to resource constraints. This period was chosen for its relevance to the early COVID-19 pandemic, as the newspaper's coverage of other countries' management of the crisis provides a context where even non-experts on China could pose interesting questions.

doi:10.6342/NTU202502354

4.1.1 Translation into English

All articles were translated into English using `gpt-4o-mini`.[1] The translation prompt is included in Appendix B.1.1. After automatic translation, Chinese characters were left in 90 texts (2.1%), all of which were manually checked and replaced by appropriate English translation where necessary. A detailed breakdown of cases where Chinese characters were left in the translations is included in Appendix B.1.2.

4.2 Architectures

This section introduces the three RAG architectures compared in this experiment, detailing the setup of the datastore (indexing) for each architecture and describing how the QA pipeline is structured, including both the retrieval and generation processes.

4.2.1 VectorRAG

Indexing

Two vector databases were created: one based on the original Chinese texts and one on the English translations. Embeddings were generated using *text-embedding-3-small*, with 1000-token chunks and a 20-token overlap. The embeddings were stored in the vector databases along with metadata linking each chunk back to the original text and its position within the text.

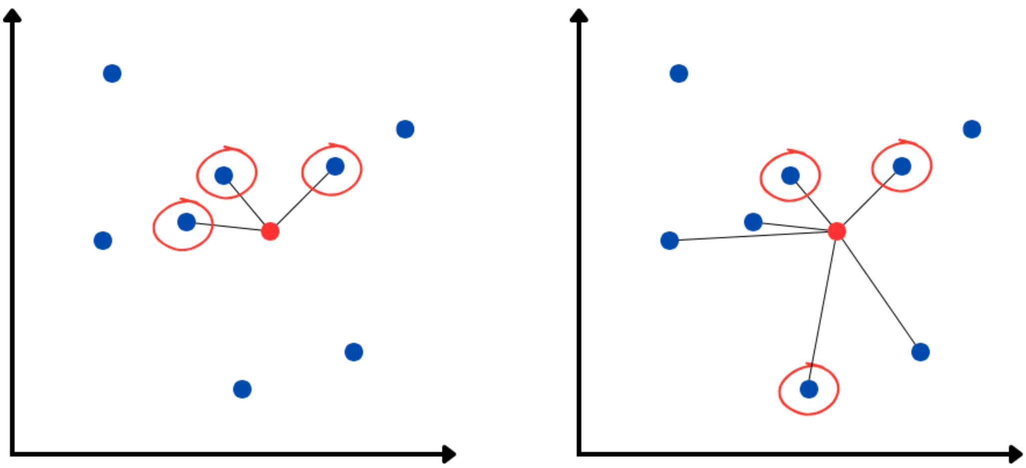
[1] The choice of model was made after initial attempts to translate individual articles using `glm-4-9b-chat` and `gemini-1.5-flash`. The former would occasionally leave individual Chinese characters untranslated in the middle of an English translation, while the latter would not respond to a number of requests for unknown reasons.

Question Answering

The VectorRAG implementation employs Maximal Marginal Relevance (MMR) for document retrieval. MMR, as introduced by Carbonell and Goldstein (1998), optimizes the trade-off between relevance and diversity when selecting documents. The algorithm selects documents iteratively by maximizing the following criterion:

$$\text{MMR} = \arg \max_{D_i \in R \setminus S} \left[\lambda(\text{Sim}_1(D_i, Q)) - (1 - \lambda) \max_{D_j \in S} \text{Sim}_2(D_i, D_j) \right] \tag{4.1}$$

Here, Q represents the query, R is the ranked list of documents by relevance, S is the subset of documents already selected, and $R \setminus S$ includes the remaining candidates. The terms Sim_1 and Sim_2 are similarity metrics, while λ (a value between 0 and 1) determines the trade-off between relevance and diversity. Values closer to 1 prioritize relevance, while values closer to 0 emphasize diversity. Figure 4.1 illustrates the algorithm. For VectorRAG, the *LangChain* implementation of the MMR retriever is used with cosine similarity for Sim_1 and Sim_2, $\lambda = 0.5$, selecting three documents from the top nine.



(a) Similarity Search (b) MMR Search

Figure 4.1: A visualization of the difference between Similarity Search and Maximal Marginal Relevance (MMR) Search. Similarity Search finds and returns the N most similar vectors, MMR Search returns N maximally diverse vectors from the set of the M most similar vectors, providing a balance between similarity and diversity.

The question answering pipeline is illustrated in Figure 4.2, with a simplified example. To match the query language, both the system instructions and prompt template are provided in English and German, following findings in Chirkova et al. (2024) that suggest this approach improves performance (see Section 2.3.2). `gpt-4o-mini` generates responses based on the retrieved context. The full set of prompts for both languages is provided in Appendix B.3.1.

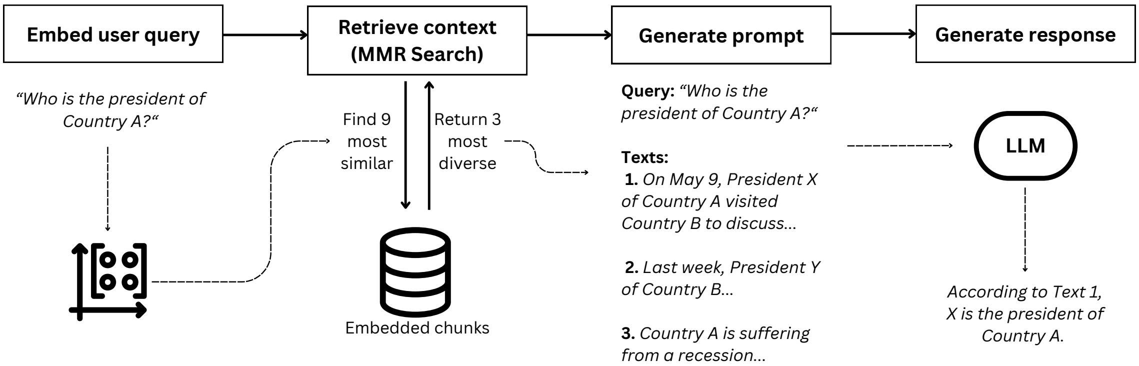


Figure 4.2: Overview of the VectorRAG architecture with MMR search. My implementation uses `gpt-4o-mini`.

4.2.2 CypherRAG

Indexing

The CypherRAG implementation utilizes `DiffbotGraphTransformer`, a *LangChain* tool, to generate a knowledge graph. During initial exploration, this tool stood out for its ability to reliably produce graphs in a format directly compatible with *Neo4j*. It also performs entity resolution, identifying and linking entities across articles by associating nodes with corresponding *Wikidata* entries (Vrandečić and Krötzsch, 2014) where possible. Additionally, the tool annotates edges with an "evidence" property that contains a text snippet which supports the existence of the relationship.

doi:10.6342/NTU202502354

During indexing with the `DiffbotGraphTransformer`, a separate `GraphDocument` (a *LangChain* data structure for graphs) is created for each article based on a pre-defined schema. These graphs are then loaded into a *Neo4j* database and merged into one large graph. Before the second step, I added the filename of each article as a property to every relationship in its graph, ensuring this information was not lost during the merging process.

Only an English version of the database was created, as the `DiffbotGraphTransformer` cannot extract relationships from Chinese texts. To ensure comprehensive extraction of relationships, the confidence threshold for extraction was set to 0.

Figure 4.3 shows a small subset of the graph database. The full graph schema, which is the default schema used by the `DiffbotGraphTransformer`, is included in Appendix B.2.

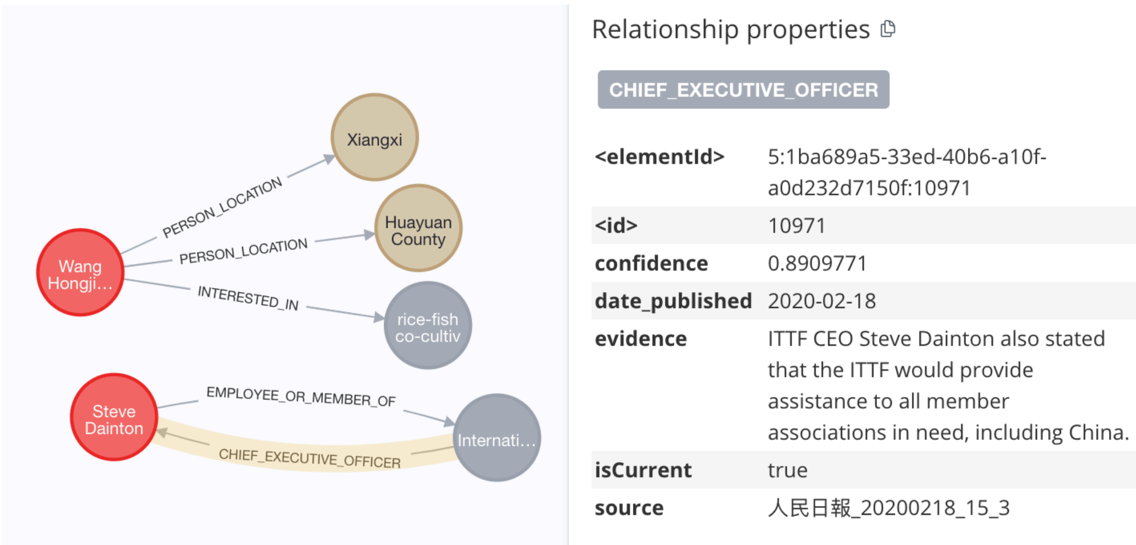


Figure 4.3: A screenshot of the view of a subset of 5 nodes of the *Neo4j* graph database in the *Neo4j* Browser. The properties of the highlighted relationship are shown on the right.

doi:10.6342/NTU202502354

Question Answering

The CypherRAG implementation uses the *LangChain* `GraphCypherQAChain` with `gpt-4o` for both retrieval and generation. The process begins by passing the user query and the graph schema to the LLM, which generates a Cypher query to search the *Neo4j* graph database. The results of the graph query are then incorporated into a second prompt as context, and `gpt-4o` generates the final answer to the user query.

The pipeline is illustrated in Figure 4.4. As with the VectorRAG implementation, the prompt language matches the target language. For German queries, the Cypher generation prompt is in English (since the entity and relationship names in the Cypher output should be in English), while the answer generation prompt is in German. For English queries, the default prompts are used. The prompts for the German version are provided in Appendix B.3.2.

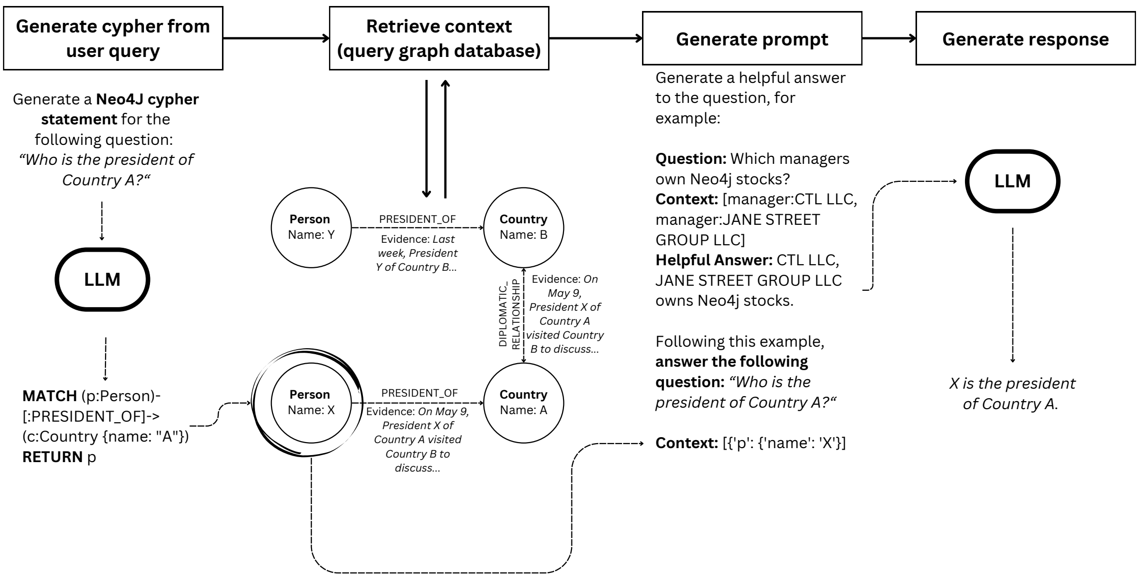


Figure 4.4: Overview of the *LangChain* `GraphCypherQAChain` (referred to as CypherRAG in this paper). Prompts are simplified for illustration purposes. My implementation uses `gpt-4o`.

doi:10.6342/NTU202502354

4.2.3 GraphRAG

Indexing

For the GraphRAG implementation, the English database was indexed using the default prompts and settings from the *GraphRAG* Python library, with only the LLM changed to `gpt-4o-mini` due to the high cost of indexing with `gpt-4o`. For the Chinese database, the indexing process used Simplified Chinese versions of the four indexing prompts[2] to align with the dataset.

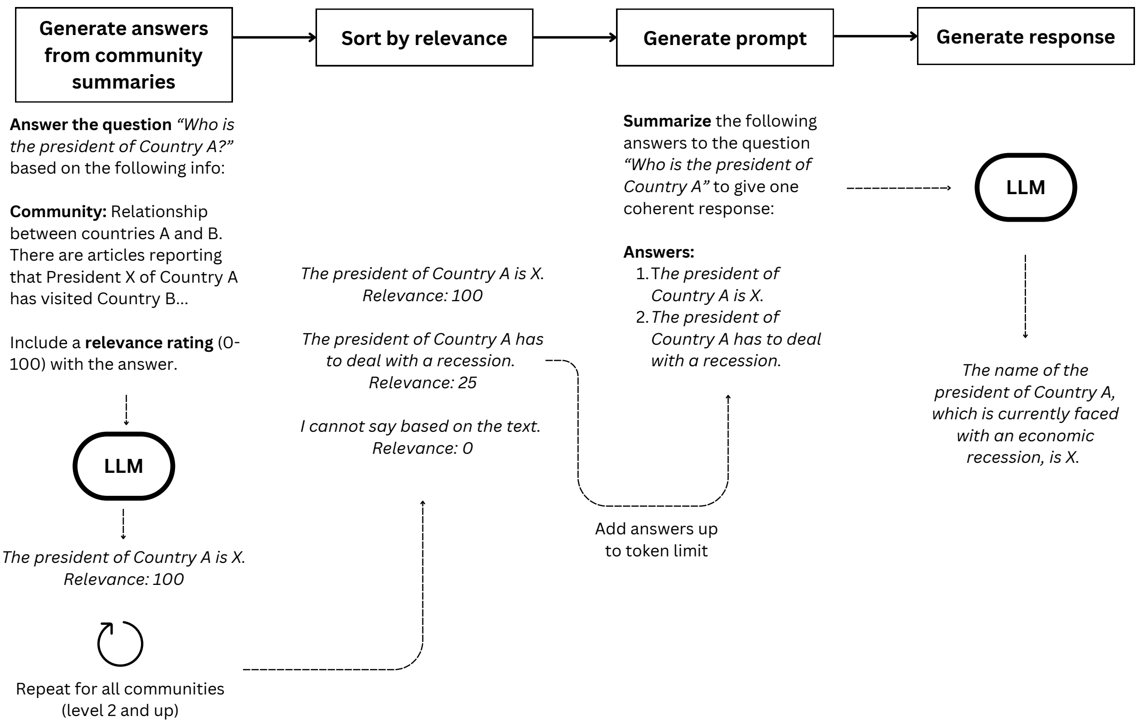


Figure 4.5: Overview of GraphRAG global search according to Edge et al. (2024). Prompts are simplified for illustration purposes. My implementation uses `gpt-4o-mini`.

[2] The Chinese translations of the indexing prompts were taken from `https://blog.csdn.net/engchina/article/details/140733785` (accessed October 2, 2024).

Question Answering

GraphRAG used the global search functionality with default settings (community level 2, response type "multiple paragraphs") and default English prompts, for both German and English queries.[3] A simplified example of the QA pipeline is shown in Figure 4.5. Global search works by generating an answer for each community summary at or above the specified community level. The answers are then ranked by relevance to the user query, and as many as possible are added to a second prompt, which directs the LLM to generate a comprehensive response based on the previous answers.

4.3 Evaluation

Many evaluation methods in the literature rely on existing open-domain QA datasets, using metrics such as the BLEU score to rate answer quality. I decided to create a custom set of questions and evaluation criteria for the following reasons:

- There are, to my knowledge, no existing QA datasets specific to the People's Daily.
- Evaluation methods should align with the purpose of the system. Since I am comparing three different architectures, the scoring criteria must be different for each.
- I preferred manual evaluation where possible, since automatic scoring against a model answer fails to account for the possibility of finding an equally correct answer elsewhere in the data.

The remainder of this section explains how the questions were created and scored.

[3] Although German prompts could have been implemented, queries had already been executed using the default English prompt. The output language was consistently correct, so rerunning the queries with modified prompts was deemed unnecessary given the high cost of each request (see Section 5.2.2).

doi:10.6342/NTU202502354

4.3.1 Question Generation

For each of the following question types, a total of 27 questions[4] were created:

Detail questions were designed to be answerable using information from a single article. A total of nine articles were selected, and for each article, an LLM was prompted to generate three fact-based questions along with model answers (see Appendix B.4.1 for the full prompt). Some questions were manually adjusted where they were deemed unsuitable for the task as annotated in the full list of questions. All questions were translated into German using ChatGPT, with minor manual edits to make the translations more natural.

Connection questions were manually written by analyzing the graph database created during the indexing of CypherRAG for relevant relationships. These questions focus on paths between two entities or multiple connections involving the same entity. The full set of questions is annotated with the distance between the nodes involved and the number of articles that the relevant subset of the graph database is based on.

Big Picture questions focus on the dataset as a whole rather than individual articles. These questions were generated following the method described in the original GraphRAG paper (Edge et al., 2024). An LLM was prompted with a description of the dataset to generate three potential users, three use cases per user, and three questions per use case (see Appendix B.4.2 for the full prompt).

[4] The full set of questions, their translations, and model-generated answers is available at:
Detail questions `https://docs.google.com/spreadsheets/d/1zoVvsKLbSaVm2Y3Gb-s1zQ9kSWT5H5AgRpiRNsV1YfQ/edit?usp=sharing`
Connection questions `https://docs.google.com/spreadsheets/d/1RuaRTwV68sC-2_RQN97R-KZgyKIGYpXbeapwG7jvyUA/edit?usp=sharing`
Big Picture questions `https://docs.google.com/spreadsheets/d/15YreZt9gSSUs-5d45F6uZab-Q9NkAgSTUZEAM0ZFCX8/edit?usp=sharing`

doi:10.6342/NTU202502354

4.3.2 Scoring Criteria

Answers to both Detail and Connection questions were scored manually[5] on a three-point scale: 1 point for correct answers matching the criteria, 0.5 points for partially correct or imprecise answers, and 0 points for incorrect or missing answers. The full scoring criteria for manual evaluation are detailed in Table 4.2.

For Big Picture questions, manual scoring was not feasible due to their more open-ended nature. Instead, I adopted the approach from Edge et al. (2024), using an LLM (`gpt-4o-mini`) to perform pairwise comparisons based on defined criteria (*Comprehensiveness*, *Empowerment*, and *Directness*; see Table 4.3) with slight modifications. While the original study used four criteria, I reduced this to three, omitting *Diversity* because its definition overlapped with *Comprehensiveness*, and their score distributions were similar in the original experiment. As in the original study, the LLM was instructed to assign 1 point to the better answer. However, I introduced the additional options of assigning 0 points to both answers (e.g., if both were "I don't know") or 1 point to both if no meaningful differences were observed. This adjustment is meant to account for cases in my experiment where the only difference between the systems was the query language, making it plausible that some answer pairs would exhibit no significant variation. The LLM-based evaluation was done twice, once with German prompts and once with English prompts (see Appendix B.5), and the scores were averaged to counter a potential language bias when an English response is compared against a German one. Section 4.6 further explores the influence of prompt language on LLM-based scoring.

[5] Despite striving for objective criteria, certain cases required subjective judgment. In edge cases, my reasoning for the scores is annotated in the "Judgment" column of the full datasets.

doi:10.6342/NTU202502354

Score	**Detail Questions Criteria**	**Connection Questions Criteria**
1 point	• Answer matches the sample answer closely (with minor deviations in details). • Answer contains the sample answer or something close, and additional information is not misleading. • Answer differs from the sample but is still correct given the retrieved context (e.g., for "What has X said about Y?", any correct statement is acceptable).	• Clear connection established (e.g., path via mutual acquaintances or common interests). • For listing questions: Does not need to be complete but must reference at least three different articles with no wrong listings (except for the Tokyo Stock Exchange question, which has only one valid result).
0.5 points	• Correct but imprecise answer (e.g., only the year is mentioned for "When did event X happen?" even if the sample answer is more precise). • Answer contains the sample answer or something close, but additional information is imprecise or misleading. • Mostly correct answer (e.g., correct entity found but misspelled).	• Vague or speculative answer. • Correct but missing details. • Connection is present, but one incorrect connection is made. • For listing questions: List is incomplete (fewer than three articles referenced) and/or includes wrong listings (no more than one-third of the total).
0 points	• No answer. • Wrong or mostly wrong answer. • Mostly wrong answer (e.g., for "Wann nahm Cai Dafeng an der Amtseinführung von Präsident Nyusi teil?", the answer "2015" is wrong because Nyusi's second inauguration, which Cai Dafeng attended, was in 2020).	• No answer. • Wrong answer. • Connection is present, but at least two incorrect connections are made. • For listing questions: More than one-third of the items are incorrect.

Table 4.2: Scoring criteria for the manual evaluation of Detail and Connection questions.

doi:10.6342/NTU202502354

Criterion	Description
Comprehensiveness	How well does the answer cover the aspects and details of the question? Does it provide relevant, complete, and detailed information?
Empowerment	How well does the answer help the reader understand the topic? Does it equip the reader to make informed judgments or decisions?
Directness	How clearly and specifically does the answer address the question? Does it avoid unnecessary digressions?

Table 4.3: Scoring Criteria for the LLM-based evaluation of Big Picture questions. Criteria are based on Edge et al. (2024).

4.4 Performance of Different Architectures Across Question Types

Answers for the 81 questions were generated using all ten setups described in Section 4.2. From here on, each setup is identified by a three-part notation indicating the system, database language, and query language:

- The first part refers to the system (*VectorRAG*, *CypherRAG*, or *GraphRAG*).
- The second part indicates the database language (*dbEN* for English and *dbZH* for Chinese).
- The third part specifies the query language (*qaEN* for English and *qaDE* for German).

doi:10.6342/NTU202502354

For example, *GraphRAG_dbZH_qaDE* refers to the GraphRAG architecture with a Chinese datastore and German queries. Note that *CypherRAG_dbZH_qaEN* and *CypherRAG_dbZH_qaDE* are not included, as no graph database could be generated from the Chinese texts.

Detail Questions

VectorRAG performed best on Detail questions, as was expected. Its scores ranged from 18.5 to 21.5, depending on the language combination. CypherRAG was unable to answer any of the questions. GraphRAG came in second, scoring between 13.5 and 15.5 points. The results are visualized in Figure 4.6

Quantitative analysis was performed using linear regression implemented with the `statsmodels` package (Seabold and Perktold, 2010). The model included the RAG system, database language, and query language as independent variables and the score on each question as the dependent variable. CypherRAG was excluded from the regres-

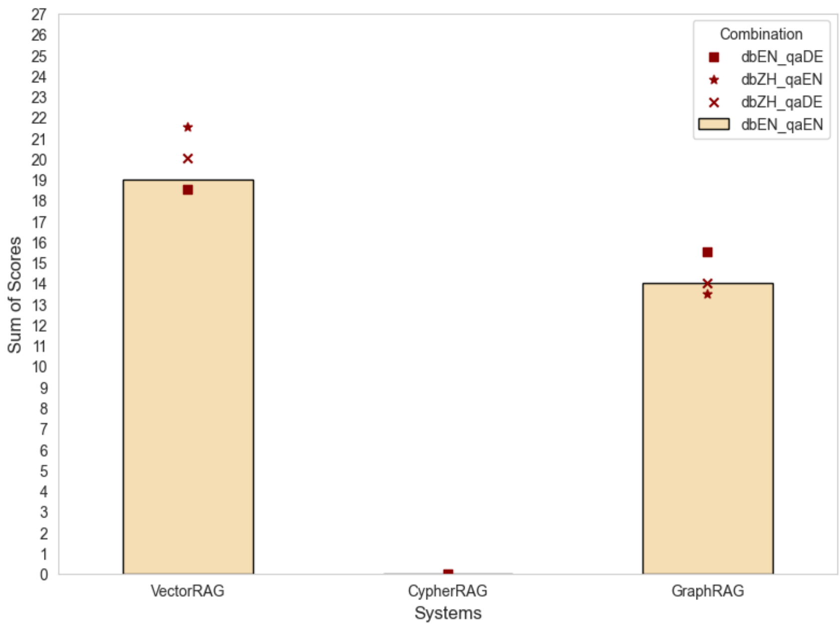


Figure 4.6: Total score achieved by each system on Detail questions.

doi:10.6342/NTU202502354

sion to prevent a group with zero variance and zero scores from skewing the model. Using VectorRAG with the English database and English queries as the baseline, the model showed that GraphRAG scored significantly lower by approximately 0.2 points per question ($\beta = -0.204$, $p = 0.001$). Neither the choice of database language (*dbZH* vs. *dbEN*) nor query language (*qaDE* vs. *qaEN*) had a statistically significant effect on scores. The overall model explained about 5% of the variance in scores ($R^2 = 0.049$, $F(3, 212) = 3.661$, $p = 0.013$).

Connection Questions

None of the systems scored more than 12.5 out of 27 points. As expected, CypherRAG achieved the highest scores, with no difference in scores between German and English queries. GraphRAG achieved 12 points with one setup (*dbEN_qaDE*). An overview of all scores is included in Figure 4.7.

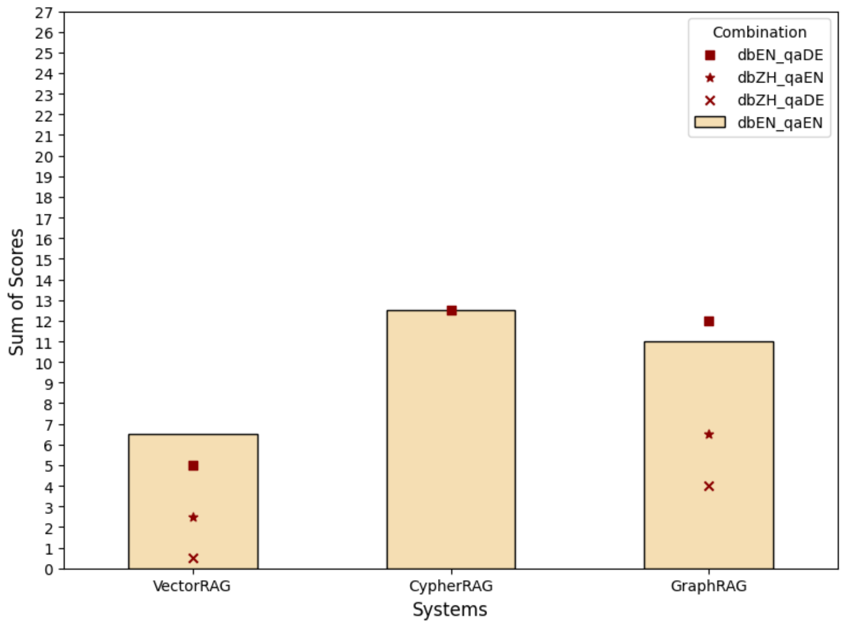


Figure 4.7: Total score achieved by each system on Connection questions.

doi:10.6342/NTU202502354

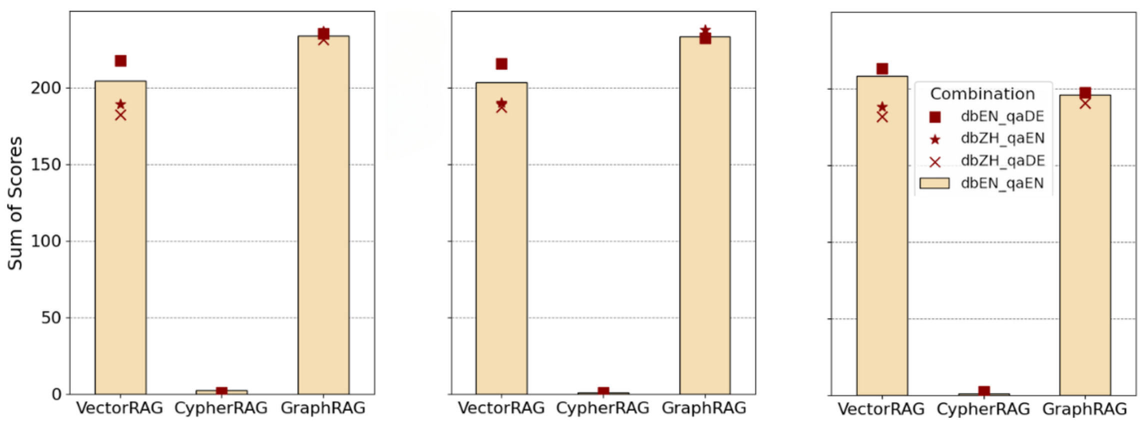
To investigate the influence of the RAG system and language variables on performance quantitatively, another linear regression analysis was conducted. CypherRAG and English (*dbEN*, *qaEN*) were set as baseline categories. The results show a significant effect of the RAG system and database language on scores ($R^2 = 0.146$, $F(4, 265) = 11.31$, $p < 0.001$). Specifically, VectorRAG scored significantly lower than CypherRAG ($\beta = -0.232$, $p = 0.001$), and the use of a Chinese database (*dbZH*) was associated with reduced scores compared to the English database ($\beta = -0.194$, $p < 0.001$). No significant effect was observed for query language (*qaDE* vs. *qaEN*) or for GraphRAG compared to CypherRAG.

A closer look at the correct answers (Score of 1) from the best-performing VectorRAG architecture (*dbEN_qaEN*) and the best-performing GraphRAG architecture (*dbEN_qaDE*) reveals a pattern. One of the metadata types included in the set of Connection questions is *number of articles*, which speficies how many articles were involved in the generation of the subgraph that was used as a basis for the question (see Section 4.3.1). The median number of articles used for question generation was higher for GraphRAG (3 articles) than for VectorRAG (1 article), with the 1st and 3rd quartiles being 2 and 6 articles for GraphRAG, and 1 and 2 articles for VectorRAG. While the answers were not necessarily always based on the same articles as the subgraph from the graph database that could have been used to answer the question, this result is in line with what would be expected under the assumption that the number of articles gives an indication of how diffused the information required to answer the question is across the dataset.

Big Picture Questions

Figure 4.8 summarizes the total summed scores by RAG type for Big Picture questions across the three scoring criteria *Comprehensiveness*, *Empowerment*, and *Directness*. GraphRAG outperformed the other systems in terms of *Comprehensiveness* and *Empowerment*, with negligible differences between the database and query languages. I again performed a linear regression for each of the criteria, with the dependent variable being the number of wins achieved by each system on each question. CypherRAG was excluded due to the negligible number of points it achieved (2.5 or fewer for every condition). Given that CypherRAG had responded to only one of the questions, this outcome of the LLM-based evaluation was expected.

The linear regression models for *Comprehensiveness* ($R^2 = 0.141$, $F(3, 212) = 11.62$, $p < 0.001$) and *Empowerment* ($R^2 = 0.114$, $F(3, 212) = 9.059$, $p < 0.001$) were statistically significant. GraphRAG served as the baseline and outperformed VectorRAG on both *Comprehensiveness* ($\beta = -1.287$, $p < 0.001$) and *Empowerment* ($\beta = -1.232$,



(a) Comprehensiveness (b) Empowerment (c) Directness

Figure 4.8: Total summed scores achieved by each system for *Comprehensiveness*, *Empowerment* and *Directness* on Big Picture questions.

doi:10.6342/NTU202502354

$p < 0.001$). The only significant effect of language choice was observed for the *Comprehensiveness* criterion, with Chinese databases performing worse than English ones ($\beta = -0.500$, $p = 0.034$).

VectorRAG performing better in terms of *Directness* is in line with expectations, as Edge et al. (2024) saw similar results with a classic vectorstore-based system (see Section 2.3.1). However, this difference did not prove significant in the linear regression model ($p = 0.583$). For *Directness*, the overall model only explained 2.4% of the variance and was not significant ($R^2 = 0.024$, $F(3, 212) = 1.745$, $p = 0.159$). Nonetheless, responses based on Chinese databases won less often than those based on English databases ($\beta = -0.556$, $p = 0.029$).

The query language was not a significant factor for any of the three criteria.

4.5 Analysis of Patterns in Answers and Mistakes

In the scoring criteria for manual evaluation (see Table 4.2), both an answer of "I don't know" and a factually incorrect answer are given 0 points. This section explores what types of mistakes commonly led to a score of 0 for each system and makes additional observations about patterns in answers.

4.5.1 VectorRAG

The VectorRAG prompts explicitly ask the LLM to answer with "none" when the texts are not helpful for generating an answer. In the answers given to both Detail and Connection questions, nearly all scores of 0 were given to "none" answers where the sys-

doi:10.6342/NTU202502354

tem answered with “none” or otherwise articulated that the retrieved context does not contain the relevant information, with only two answers to Connection questions receiving a score of 0 for being factually incorrect.

Cases where scores of 0.5 were given sometimes contained smaller inaccuracies or half-truths, such as the following answer generated by *VectorRAG_dbEN_qaEN*:

Question: What are some key characteristics of the Party’s approach to self-revolution and internal reflection as described by Zhang Jinghua?

Expected Answer: To achieve this, we must continue to **work hard on strict governance** of the Party, focus on the key minority and **core positions**, strengthen **daily supervision**, and timely ‘redden faces and make them sweat’ ensuring that Party members and cadres do not lose their **original aspiration** and always **remember their roots.**

Given Answer: Zhang Jinghua’s description of the Party’s approach to self-revolution and internal reflection highlights several key characteristics:
Zhang Jinghua’s description of the Party’s approach to self-revolution and internal reflection highlights several key characteristics:
1. **Courage to Confront Problems**: The Party emphasizes the importance of facing issues head-on and the willingness to correct mistakes. This self-awareness and introspection are seen as vital for maintaining the Party’s integrity and effectiveness.
2. **Self-Criticism and Reflection**: Party members are encouraged to engage in self-questioning and reflection regularly. This practice is intended to awaken their

doi:10.6342/NTU202502354

minds, clarify their vision, and ensure that their original intentions remain intact.

...

4. **Promotion of a Spirit of Struggle**: The Party's history is marked by struggle

...

Although the answer included several points that were put forward by Zhang Jinghua, other points were not made by him (a keyword search to find other mentions of Zhang Jinghua returned empty results). His name was also absent from the retrieved texts, meaning the mostly correct response was coincidental. This example demonstrates that even instructing the LLM to respond only when confident does not entirely prevent errors.

4.5.2 CypherRAG

CypherRAG is a special case as its retrieval relies on LLM-generated code being correct and there are multiple places in the pipeline where the answering process can fail.

For Detail questions, the total score of 0 was due to the system responding with "I don't know the answer" to every single question.

Mistakes in answers to Connection questions were more diverse in nature. Figure 4.9 details the results of a separate assessment of retrieval and generation for each answered Connection question according to the following criteria:

- **Retrieval Correct** - Is the generated Cypher able to retrieve relevant information from the graph?

doi:10.6342/NTU202502354

- **Retrieval Complete** - Does the generated Cypher retrieve *all* relevant information?
- **Generation Correct** - Based on retrieved information, are the details given in the answer correct?
- **Generation Complete** - Does the answer contain *all* relevant information contained in the retrieved context?

Generation was only assessed where retrieval succeeded.

Incorrect retrieval occurred for different reasons (including the number of occurrences by query language):

- The pipeline stopped due to syntax issues in the generated Cypher, causing the call to the database to fail (qaEN 5, qaDE 3).
- The generated Cypher was inefficient and took an unacceptable amount of time to run, e.g. by trying to find all paths rather than only the shortest one (qaEN 2, qaDE 1).
- Retrieval yielded empty, irrelevant or insufficient context to generate an answer (qaEN 6, qaDE 9).

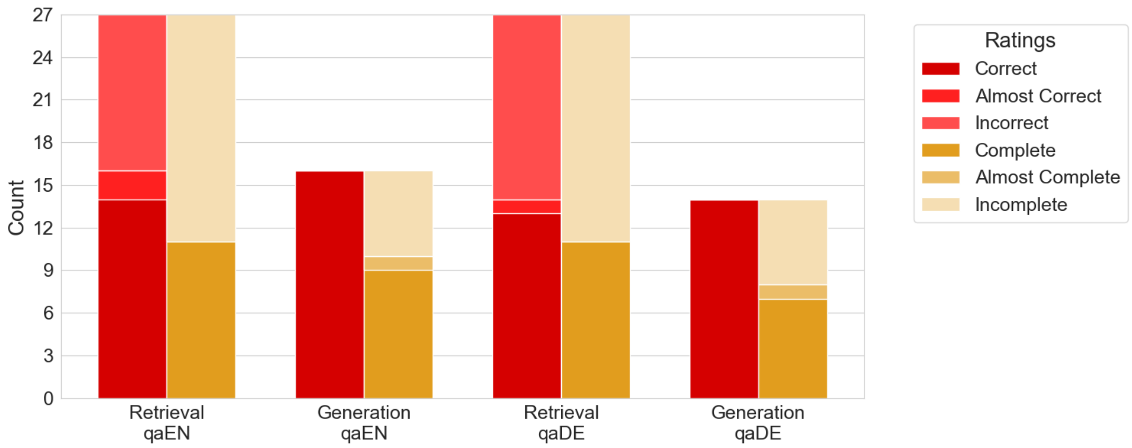


Figure 4.9: Ratings of retrieval and generation outputs generated by CypherRAG.

doi:10.6342/NTU202502354

One contributing factor to the slightly higher number of incorrect retrievals with German queries was the translation of entity names. This is because Cypher queries return nodes from the database only if the name given is an exact match. As a result, the first prompt must translate the German entity name back into English if the German name differs. An example where this worked well is “Renmin-Universität China.” The returned Cypher was

```
(o:Organization {name: "Renmin University of China"}),
```

which is the name the university has in the database. “Mitgliedern des Französisch-Chinesischen Komitees,” however, returned

```
(p:Person)-[:EMPLOYEE_OR_MEMBER_OF]
->(org:Organization {name: "French-Chinese Committee"}).
```

While not wrong, the entity in question is called “Franco-Chinese Committee” in the database, leading to nothing being returned and leaving the second prompt without context. The same issue could also occur with English prompts if a synonym is used.

An example for correct, but incomplete retrieval is the Cypher generated by *Cypher-RAG_dbEN_qaEN* for the question “Can you find any person born after 1950 who is a parent? Please provide evidence and sources.”:

```
MATCH (p:Person)-[r:HAS_CHILD]->(:Person)
WHERE p.dateOfBirth > '1950'
RETURN p.name AS personName, p.dateOfBirth AS dateOfBirth,
r.evidence AS evidence, r.source AS source
```


doi:10.6342/NTU202502354

While this Cypher is able to retrieve ‘Person’ nodes with a ‘HAS_CHILD’ relationship, it does not account for the possibility that a person’s child is not listed in a node in the graph. Instead, the parental status of a person could be annotated via the ‘numberOfChildren’ attribute.

During the generation step, there were no problems with correctness. Where sufficient context was retrieved, the content of answers was strictly based on the returned context. There were, however, a few instances of insufficient explanation in the answer, leading to generation being rated as incomplete.

The following is such an example (generated by *CypherRAG_dbEN_qaEN*):

Question: How many employees of Renmin University of China do you know of, and what evidence and sources can you provide for their employment at the university?

Retrieved Context: see Table 4.4

Given Answer: I know of 8 employees of Renmin University of China. The evidence and sources for their employment are as follows:

1. Liu Zhenyun, Professor at Renmin University of China (Source: 人民日報_20200226_20_1).

2. Tang Jun, Associate Professor at the School of Public Administration at Renmin University of China (Source: 人民日報_20200224_7_1).

3. Gao Mingxuan, Law Professor at Renmin University of China (Source: 人民日報_20200123_1_7).

4. Zheng Gongcheng, Professor at Renmin University of China and President of the China Social Security Society (Source: 人民日報_20200117_7_2).

doi:10.6342/NTU202502354

5. Yang Weiguo, Dean of the School of Labor and Human Resources at Renmin University of China (Source: 人民日報 _20200107_7_1).

6. Zhou Guangsuo, Associate Professor at the School of Labor and Human Resources at Renmin University of China (Source: 人民日報 _20200107_7_1).

7. Wang Sangui, Director of the China Poverty Research Institute at Renmin University of China (Source: 人民日報 _20200228_18_1).

The generated answer was rated as correct given the retrieved evidence since the first column of the evidence was correctly summed (“8 people”), and the LLM recognized that Wang Huning, one of the names retrieved, was not mentioned as an employee but a visitor. However, the answer proceeded to list only seven names without elaborating on the apparent contradiction, which is why the generation was rated as incomplete in this case.

doi:10.6342/NTU202502354

numberOfEmployees	evidence	source
1	"**Liu Zhenyun** (Professor at Renmin University of China)"	人民日報 _20200226_20_1
1	"**Tang Jun,** an associate professor at the School of Public Administration at Renmin University of China, indicated that studying public health safety involves not only medicine and health sciences but also integrating theories and methods from various disciplines such as sociology, politics, public security, psychology, and ecology."	人民日報 _20200224_7_1
2	"When visiting **Gao Mingxuan**, a law professor at Renmin University of China and a recipient of the national honorary title of 'People's Educator,' **Wang Huning** inquired about his health and living conditions and listened to his suggestions for constructing a philosophy and social sciences with Chinese characteristics."	人民日報 _20200123_1_7
1	"**Zheng Gongcheng**, a professor at Renmin University of China and president of the China Social Security Society, stated that China has now built the world's largest medical security system, benefiting approximately 19% of the global population, creating a miracle in the history of human development in terms of medical insurance reform and development."	人民日報 _20200117_7_2
1	"As online positions continue to increase and the flexible employment group rapidly expands, **Yang Weiguo**, dean of the School of Labor and Human Resources at Renmin University of China, pointed out that how to protect the legal rights of these workers, including reasonable wage levels and working hours, is becoming a new issue."	人民日報 _20200107_7_1
1	"**Zhou Guangsuo**, an associate professor at the School of Labor and Human Resources at Renmin University of China, stated that the digital transformation of the tertiary industry has given rise to new models such as platform economy and sharing economy, with the platform economy being particularly prominent, becoming a new engine driving entrepreneurship and employment in recent years."	人民日報 _20200107_7_1
1	"**Wang Sangui**, director of the China Poverty Research Institute at Renmin University of China, suggests sorting out existing poverty alleviation policies and further strengthening policy measures conducive to stable poverty alleviation, including health poverty alleviation, skills training, and dual support for 'will and wisdom.'"	人民日報 _20200228_18_1

Table 4.4: Context retrieved by *CypherRAG_dbEN_qaEN* when asked to find the number of employees of Renmin University of China.

doi:10.6342/NTU202502354

From the Big Picture set, only one question was answered (generated by *CypherRAG_dbEN_qaEN*, with a similar answer given by *CypherRAG_dbEN_qaDE*):

Question: Can we identify any cultural trends or movements that gained prominence during this period?

Retrieved Context:

s.name	numberOfPeopleInterested
"public security"	13
"traditional Chinese medicine"	7
"art of painting"	5
"intensive care medicine"	5
...	...

Given Answer: Public security, traditional Chinese medicine, and the art of painting gained prominence during this period, with 13, 7, and 5 people interested, respectively.

This seemingly strange answer can be attributed to CypherRAG making an effort to construct a Cypher query for any question (in fact, only a single instance of no Cypher being generated was observed, which was for a Detail question).

In the above example, the generated Cypher was

```
MATCH (p:Person)-[:INTERESTED_IN]->(s:Skill)
RETURN s.name, COUNT(p) AS numberOfPeopleInterested
ORDER BY numberOfPeopleInterested DESC
```


doi:10.6342/NTU202502354

The full list of generated Cypher statements is included in the QA datasets.

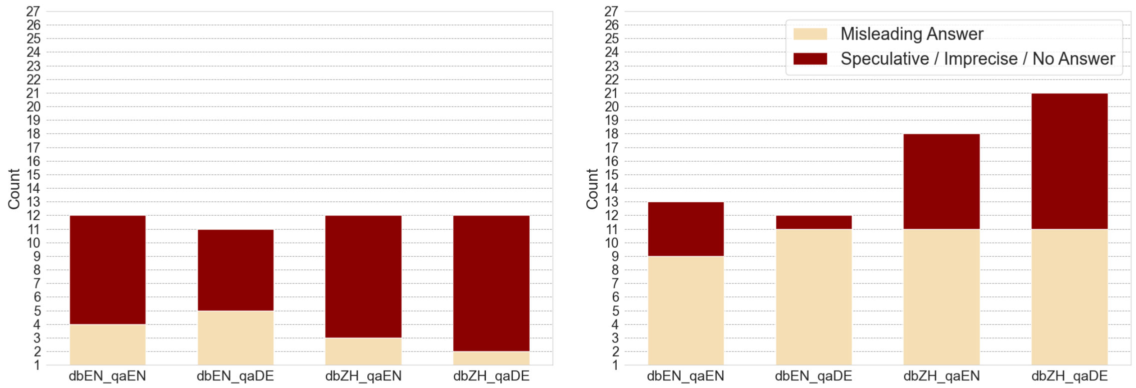
4.5.3 GraphRAG

While it is difficult to assess the correctness of GraphRAG's answers to Big Picture questions, it is possible to do so for Detail and Connection Questions.

As shown in Figure 4.10, misleading answers were common, especially for Connection questions. This is different from VectorRAG and CypherRAG, which both had a low incidence of misleading answers.

The results show that despite the GraphRAG generation prompt explicitly stating not to make anything up, this still happens in many cases where information is insufficient.

Pinpointing the exact cause for each individual misleading answer was not feasible due to mistakes in the community report numbers cited es evidence in the answers. During manual evaluation, I noticed that many GraphRAG answers ended with "[Data: Reports (1, 2, 3, 4, 5, +more)]" or would list the same community report number several times. In other cases, nonexistent community reports were cited. For example, when a report



(a) Detail questions (b) Connection questions

Figure 4.10: Number of answers generated by GraphRAG that received a score of 0, along with the proportion of answers that were misleading vs. answers where the system did not give a straight answer or admitted to not knowing.

doi:10.6342/NTU202502354

was cited correctly, if it contained numbers referring to lower-level parts of the graph, those numbers would also be added to the final answer as a supposed "Community Report number."

In order to still perform the manual evaluation when an answer differed from those evaluated previously by CypherRAG and VectorRAG, I resorted to performing keyword searches of entities mentioned in the answer and reading the original articles to determine the correctness of the claims.

4.6 Additional Analysis: Language Bias in LLM-Based Evaluation

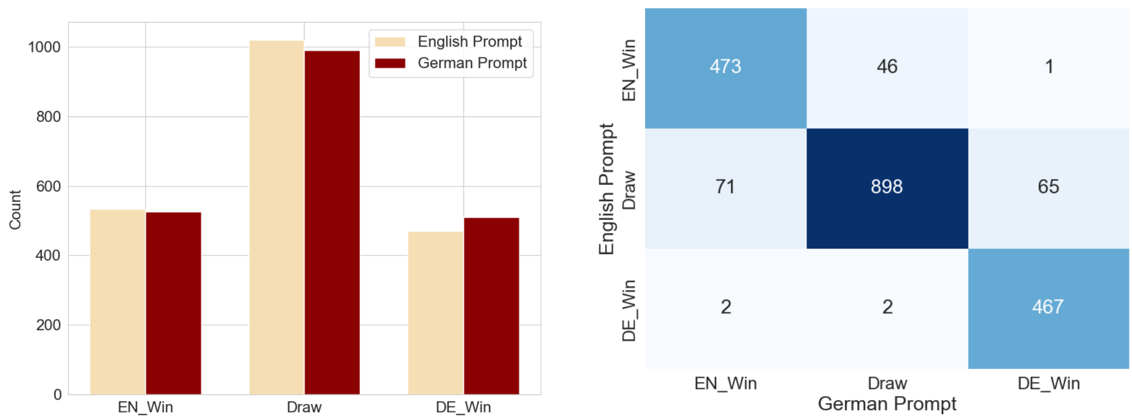
To assess whether the language of the prompt asking the LLM to compare two answers makes a difference to the results, I filtered the data to only those cases where a German answer was directly compared to an English answer and categorized answers by *EN_Win* (cases where the LLM preferred the English answer), *DE_Win* (cases where the LLM preferred the German answer) and *Draw* (both systems received the same score, including scores of (0,0) and (1,1)).

A Weighted Cohen's Kappa (Quadratic) of 0.89 between the judges indicates a high level of agreement in their assessments.

Nevertheless, Figure 4.11a shows that while the numbers between German and English prompts are similar overall, English prompts led to a slightly higher number of wins for English answers and draws, while German prompts led to a slightly higher number of wins for German answers. Figure 4.11b shows that most of the disagreements were

doi:10.6342/NTU202502354

between a draw and a win, with only three instances of opposite results. The entries above the diagonal represent same-language favor, i.e., the cases where the judge rated the answer in its own language better than the other judge did. Conversely, the entries below the diagonal represent opposite-language favor.



(a) Total Counts (b) Confusion Matrix

Figure 4.11: Counts of decisions made by `gpt-4o-mini` on direct comparisons between English and German answers with English vs. German comparison prompts.

To identify the magnitude of same-language bias, I investigated the difference between same-language and opposite-language favor:

$$\frac{\text{\# of same-language favor - \# of opposite-language favor}}{\text{\# of observations}}, \tag{4.2}$$

which yielded a same-language bias of $1.8\% \pm 0.7\%$ overall, where the uncertainty represents the standard error of the mean.[6] Split by criterion, the biases were $2.2\% \pm 0.8\%$ for *Comprehensiveness*, $4.1\% \pm 0.8\%$ for *Empowerment* and $-0.8\% \pm 1.6\%$ for *Directness*. Bias was therefore strongest for comparisons by *Empowerment*, while there was no statistically significant bias for comparisons by *Directness*.

[6]Every answer is interpreted as a single realization of an experiment; every one of these experiments yields a bias of either +1 (if the answers are same-language favored), 0 (if they agree) or -1 (opposite-language favor); the uncertainty given is then the standard error of the mean over all of these realizations.

doi:10.6342/NTU202502354

4.7 Summary of Findings

1. As expected, VectorRAG performed best on Detail questions. It also achieved a higher score than GraphRAG on Big Picture questions when evaluated for the *Directness* criterion (although not statistically significant), with reasonable performance on the *Comprehensiveness* and *Empowerment* criteria. Connection questions proved to be more difficult for VectorRAG, especially those where the path between two entities spans multiple articles.

2. CypherRAG performed best on Connection questions, while it was able to answer nearly no questions of the other types. Most problems occurred during the retrieval step.

3. GraphRAG performed best on Big Picture questions, achieving higher scores than the other setups on the *Comprehensiveness* and *Empowerment* criteria. Its biggest disadvantage when compared to the other two architectures was the high number of misleading answers given.

4. The database language and query language had a much smaller effect on scores than the RAG architecture chosen. Differences were not consistent across architectures despite `gpt-4o` and `gpt-4o-mini` being the only two LLMs used.

5. The LLM-based evaluation was shown to have a slight language bias, with `gpt-4o` being slightly more likely to favor a response in the same language as the prompt over a response in a different language.

4.8 Final Tools

For each RAG architecture, the datastores generated for the January/February 2020 dataset are made available on Github, along with instructions for querying.[7]

To use the tools on different datasets, such as the subset of articles returned by the sentiment analysis tool, indexing must be performed first. Relevant instructions are included in the repository.

[7] The RAG demonstrations are available at `https://github.com/deborahwatty/peoples_daily_rag_demo`.

doi:10.6342/NTU202502354

5 Discussion

5.1 The Case for Manual Evaluation

In the RAG experiment, I opted for manual evaluation whenever feasible. Although this approach is less common in the literature and has notable drawbacks, such as subjectivity and the limited number of test questions due to time constraints, it proved invaluable. The manual evaluation of Detail and Connection questions confirmed my concerns about the reliability of automatic evaluation methods that rely solely on gold standard answers.

My primary concern about automatic comparison against a gold standard answer was that a different, yet equally correct, answer might be marked incorrect.

As outlined in Section 4.3.1, each Detail question in my test set was generated from a single text in the corpus, linking the question to a "default" text that, when retrieved, should enable the system to produce a correct answer. Table 5.1 shows that for all Vector-RAG configurations, the number of questions for which the default article was retrieved is much lower than the total score achieved. At the same time, answers generated from other texts contribute significantly to the overall scores.

doi:10.6342/NTU202502354

System	# of "correct" articles	Score	Points from "wrong" retrieval
VectorRAG_dbEN_qaEN	11	19	8
VectorRAG_dbEN_qaDE	9	18.5	10.5
VectorRAG_dbZH_qaEN	11	21.5	11.5
VectorRAG_dbZH_qaDE	10	20	11

Table 5.1: Number of "correctly" retrieved articles (out of 27), score achieved, and "wrong" articles that still led to correct answers by VectorRAG.

When the default text is not retrieved but the answer is still correct, the two possibilities are:

1. The same information was found in a different article.

2. The answer, while different from the default, is still correct.

A closer look at the data provides examples of both. Consider this example of the first case (generated by *VectorRAG_dbEN_qaEN*):

Question: When did Cai Dafeng attend the inauguration ceremony of President Nyusi?

Expected Answer: January 15

Given Answer: Cai Dafeng attended the inauguration ceremony of President Nyusi of Mozambique on January 15.

Here, the state visit was mentioned in a different retrieved article, so the LLM could still give the expected answer. Now, consider this example of the second case (generated by *VectorRAG_dbZH_qaEN*):

Question: What is an example of a hospital that treated Covid-19 patients in Wuhan during the early phases of the outbreak?

doi:10.6342/NTU202502354

Expected Answer: Wuhan University People's Hospital, Wuhan Fifth Hospital, Wuhan Pulmonary Hospital, Hubei Armed Police Corps Hospital

Given Answer: An example of a hospital that treated Covid-19 patients in Wuhan during the early phases of the outbreak is the "Huangshan Sports Center" (洪山体育馆), which was converted into a makeshift hospital (方舱医院) to accommodate confirmed mild cases. Additionally, "Tongji Hospital" (同济医院) and "Wuhan University People's Hospital" (武汉大学人民医院) were also mentioned as facilities that treated severe cases.

Aside from the Pinyin transcription error ("Huangshan" instead of "Hongshan"), the answer is still correct based on the cited data, even though it mentions only one of the hospital names given in the default dataset. Such examples underscore the importance of using questions that have a single correct answer when using automatic scoring, which may be difficult when one does not know all the information contained in the dataset.

My second concern was the possibility of misleading additional content in the answer, which was particularly prevalent in GraphRAG answers, as shown in Section 4.5.3. One such such example was generated by *GraphRAG_dbEN_qaDE*:

Question: Wie viele neue Soldaten haben kürzlich an Fallschirmspringübungen teilgenommen?
(**Translation:** How many new soldiers have recently participated in parachute drills?)

Expected Answer: Thousands

Given Answer: (...) Kürzlich haben neue Rekruten an Fallschirmspringübungen

doi:10.6342/NTU202502354

teilgenommen, die Teil der Chadao Border Defense Community sind (...)

(**Translation:** (...) Recently, new recruits who are part of the Chadao Border Defense Community have participated in parachute drills (...))

Irrespective of whether the rest of the answer contains the correct number, if this level of detail is given, it should be correct. During manual evaluation, I performed a keyword search for “Chadao Border Defense”, which yielded one text containing the following key passage:

> *As soon as **the airplane door opened**, a powerful cold air rushed in, and snowflakes outside were swirling in the fierce wind. Against the backdrop of white snow, the words “Fuyuan Eastern Extreme Airport” stood out prominently.*
>
> *Fuyuan, located in the northeastern part of Heilongjiang Province, is the easternmost point of the motherland, known as the “Eastern Extreme.” The **soldiers** of the **Eighth Chadao Border Defense Company** stationed nearby patrol daily amidst ice and snow, guarding the safety of the motherland’s territory against the cold winds.*

According to this passage, the soldiers are patrolling, not participating in parachute drills, making that part of the answer incorrect. The first sentence gives a clue as to how this answer may have come about: During GraphRAG indexing, the Chadao Border Defense Community may have been linked to parachute drills due to its proximity to “the airplane door opened.”

doi:10.6342/NTU202502354

To summarize, manual scoring can be the right choice if:

- The test set contains questions which could possibly have correct answers other than the known ones,
- the correctness of details contained in other parts of the answer is important,
- or the goal is to gain a deeper understanding of error patterns

5.2 Limitations

This section highlights the key limitations of the experiments presented in this paper. The discussion begins with a brief analysis of the limitations specific to the Sentiment Analysis experiment, followed by a section on the more numerous limitations of the RAG experiment. These limitations provide insight into areas for improvement.

5.2.1 Sentiment Analysis

Gold Standard Creation

A single gold label was assigned based on the most frequent annotator response. This simplifies evaluation but loses information about how strong the consensus was and ignores the distribution of other responses. Despite this, gold labels were used for practicality, as alternative approaches also have limitations. Following the method in Rønningstad et al. (2024) also provides some comparability with prior work.

Choice of Prompting Strategy

Although Rønningstad et al. (2024) found few-shot prompting to be more effective, I opted for zero-shot prompting due to the length of the texts in the People's Daily Corpus. Rønningstad et al. (2024) focused on shorter texts, and it was unclear whether their findings would generalize to longer texts. While using a small number of articles from the gold dataset for few-shot prompting was an option, I chose not to for two reasons: (1) some of the articles in the gold standard dataset are very long, and (2) in the final tool, any text, including the longest ones in the entire corpus, can be selected for analysis.

Time Series Analysis

My application of time-series analysis was relatively limited, as it primarily aimed to confirm that specific anniversaries influenced the results in the anticipated direction. This was done to demonstrate that, despite fairly low inter-annotator agreement, overall trends could still be observed in a large dataset. I did not aim to identify the best model or parameters, and the model I trained included only a small subset of relevant anniversaries.

5.2.2 RAG Evaluation

For each RAG architecture, I used standard architectures, prompts, and parameters, with the only modification being the translation of prompts to match the input language. This was a deliberate choice to prioritize the comparison of multiple architectures within the available timeframe rather than tweaking the parameters of a single system. Targeted modifications that are likely to considerably improve scores for each system are discussed in Section 5.3.

doi:10.6342/NTU202502354

English Translations

While I addressed instances where LLM-generated English translations used in the RAG experiment still contained Chinese characters (see Section 4.1.1), the large size of the dataset made it impractical to thoroughly proofread all translations. Consequently, some errors or instances of unnatural phrasing may have remained, which could subtly influence the performance of retrieval and generation processes.

The quality of translations produced by `gpt-4` has been evaluated by Yan et al. (2024), with results showing performance slightly below that of a junior-level human translator.

A potential solution for future work, particularly for improving unnatural phrasing, could involve asking the LLM to rewrite the translations to better align with native speaker style, as demonstrated by Kunilovskaya et al. (2024) on human-written translations.

Indexing

My evaluation did not include a specific assessment of the indexing results. The datastores of both CypherRAG and GraphRAG are LLM-generated, and due to their large size, a comprehensive evaluation of the resulting knowledge graphs was not feasible.

Figure 5.1 illustrates one of the more glaring errors that occurred during indexing for CypherRAG. In this example, Peng Liyuan, the wife of Chinese President Xi Jinping, is incorrectly listed as also being married to Vice Premier Liu He and US President Donald Trump. The article cited for these erroneous relationships discusses a meeting between Liu He and Donald Trump, where Liu He conveys a message from Xi Jinping:

doi:10.6342/NTU202502354

My wife Peng Liyuan and I wish President Trump, Mrs. Melania, and their families a happy New Year and all the best.

Here, the LLM seems to have struggled with resolving "my" as referring to Xi Jinping, leading to the error. The generation step can, to some extent, mitigate such errors with real-world knowledge, as evidenced by what happens when we ask a related question:

Question: Who is married to Peng Liyuan?

Retrieved Context: ['spouse.name:' 'Xi Jinping' , 'spouse.name' : 'Donald Trump' , 'spouse.name' : 'Liu He']

Given Answer: Xi Jinping is married to Peng Liyuan.

Here, despite retrieving incorrect context, the model correctly identifies Xi Jinping as Peng Liyuan's spouse. Similarly, the "evidence" property in the retrieved text can sometimes help resolve such issues during the generation step, as demonstrated in the

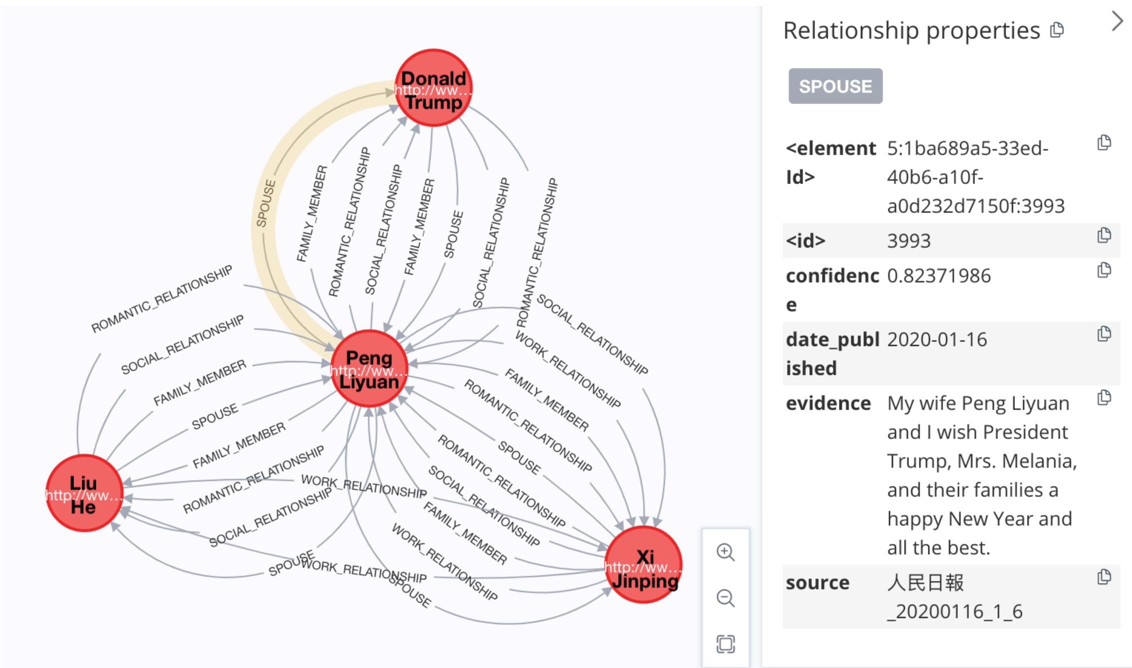


Figure 5.1: Example of an indexing mistake: Peng Liyuan is listed as having three spouses.

doi:10.6342/NTU202502354

response to the question “How many employees of Renmin University of China do you know of[...]” (see Section 4.5.2).

Nevertheless, a separate assessment of the generated knowledge graph would be ideal, which remains challenging for graphs of substantial size.

Prompt Language

In nearly all cases, I matched the prompt language to the content language, both for indexing and querying, as had been found to be helpful by Chirkova et al. (2024). However, an exception occurred with GraphRAG, where English prompts were used for German questions during the generation step. It is unclear whether this had an effect on answer quality.

At the same time, while unintended, this mismatch revealed an interesting finding. Chirkova et al. (2024), who had used smaller LLMs, emphasized the importance of aligning the prompt and content languages to ensure that the output is generated in the target language. Contrary to their findings, I observed no issues with the output language. GraphRAG consistently gave German answers to German queries even with English prompts. This suggests that the use of a larger model, such as *gpt-4o-mini*, effectively addresses this problem, mitigating concerns about the importance of prompt and content language alignment.

Evaluation of Big Picture Questions

A significant limitation of this evaluation is the inability to assess the correctness of answers to Big Picture questions. While GraphRAG performed well on metrics such as

Comprehensiveness and *Empowerment*, the high incidence of misleading answers identified during manual evaluation of other question types (see Figure 4.10) raises concerns about the accuracy of its Big Picture answers. Users may prioritize correctness over these metrics, making this a critical area of uncertainty.

The open-ended nature of Big Picture questions may have reduced the need for GraphRAG to retrieve highly specific details, potentially leading to fewer misleading answers. Since GraphRAG outputs multiple-paragraph answers by default, the system may have been compelled fill the rest of the answer with speculation for Detail and Connection questions which can often be answered fully in a single sentence. Based on this, it seems reasonable to expect that Big Picture questions, being open-ended, might result in fewer misleading answers. However, the correctness of Big Picture answers still remains unverified, highlighting a substantial limitation.

Another limitation is the reliance on an LLM to perform the pairwise evaluation. While language bias was mitigated by averaging results from English-based and German-based evaluations (see Section 4.3.2), the decision-making process of the LLM remains opaque. In one instance, a score of (3,3), an output not within the predefined range, was assigned to an answer pair, underscoring the lack of transparency.

5.3 Potential Improvements to RAG Architectures

VectorRAG might achieve higher scores on Connection questions with the inclusion of a query rewriting step. In this approach, instead of embedding the user query directly, it is first automatically transformed into one or more queries designed to produce more relevant results. Variants of query rewriting have been proposed by various authors. For

doi:10.6342/NTU202502354

example, Gao et al. (2022) generate hypothetical answers and perform retrieval using embeddings of those hypothetical answers. The rationale is that the embedding of a hypothetical answer may be more similar to that of an actual answer than that of the original question. For more reliable multi-hop question answering, Zhang et al. (2024) propose generating sub-questions, which are answered separately before being combined into a final response.

For CypherRAG, most issues occur during the retrieval step (see Section 4.5.2). When syntactically incorrect Cypher queries were generated during the experiment, asking ChatGPT to correct the query based on the error message was usually sufficient to obtain a syntactically correct Cypher. This process could be integrated into the pipeline: If the database returns a syntax error, the faulty Cypher query and the error message could be added to a prompt for an LLM to correct. The process would repeat until the error is resolved or a specified number of iterations is reached. To address empty retrievals, an initial prompt could generate lists of synonyms for the entity names in the query, improving the chances of successful retrieval.

To enhance the performance of GraphRAG, it may be beneficial to tailor retrieval and generation parameters to the question type.[1] For instance, alongside Global Search, which retrieves only community summaries, the GraphRAG package includes a Local Search option that takes the underlying knowledge graph into account. For generation, the *response_type* parameter allows users to specify the desired output length and style, which is incorporated into the prompt. By default, this is set to "multiple paragraphs", which may have contributed to the high number of misleading answers observed in the experiment, as this format can encourage over-elaboration on simple facts.

[1]Described in detail in the GraphRAG manual: `https://microsoft.github.io/graphrag/`.

doi:10.6342/NTU202502354

5.4 Potential Future Research Directions

In future work, a potential direction could involve replicating something similar to the Policy Change Index (Chan and Zhong, 2019, see Section 2.1) using sentiment analysis. By training a model on sentiment data related to a specific entity, it might be possible to detect patterns of increased positive or negative reporting before a major policy change regarding that entity is announced. This approach could help identify early signals of upcoming policy shifts based on media coverage.

Another potential research direction could examine the effect of query language on retrieval results. In my dataset, using VectorRAG across 81 questions (spanning all question types), the English database retrieved chunks mentioning "Germany" 21 times for English queries and 23 times for German queries. For the Chinese database, the numbers were 15 for English queries and 20 for German queries. While the sample size is too small to draw conclusions, these observations raise the question of whether query language influences retrieval outcomes.

Finally, combining the architectures explored in this thesis into an agentive application may be a promising direction. Such a system could be designed to answer a wide range of questions by calling different tools as needed, such as a sentiment analysis module that uses zero-shot prompting to extract entity-level sentiment over time, or a question answering module that selects among multiple RAG architectures. One option is to classify the user question and route it to the most suitable architecture; another is to generate responses using all three and let an LLM synthesize a final answer. More capabilities could be integrated as additional modular tools as needed.

doi:10.6342/NTU202502354

5.5 Conclusion

The goal of this thesis was to explore the use of LLMs for multilingual opinion mining by applying various approaches to the People's Daily newspaper. A review of relevant literature highlighted a wide range of techniques, but few have been tailored to multilingual contexts.

An initial experiment demonstrated that zero-shot prompting effectively identifies sentiment toward entities over time, with no clear advantage to using prompts written in the same language as the data. Plots of sentiment over time toward Japan in the People's Daily displayed expected patterns, such as a major shift around the time when diplomatic relations between China and Japan were normalized.

Recognizing that sentiment data alone cannot fully capture the content of the texts, a second experiment investigated the use of RAG architectures for multilingual question answering. Results showed that the choice of RAG architecture is more critical to performance than the alignment of data and query language. If the goal is to search for specific quotes, facts and numbers mentioned in the data, a vectorstore-based RAG implementation is a good option. For multi-hop questions, a RAG architecture that queries a knowledge graph during retrieval can be highly effective, but the advantages of this approach are specific to questions about entities and their relationships. Success with this method depends heavily on the comprehensive and accurate extraction of entities and connections during the indexing process. When the focus is on identifying patterns or trends across the entire dataset, a GraphRAG architecture may offer the best results. However, users should be aware that GraphRAG can sometimes produce misleading answers, necessitating careful interpretation of its outputs.

doi:10.6342/NTU202502354

An additional analysis of an LLM-based evaluation method revealed potential language biases. The content language influenced the ratings assigned by the LLM, emphasizing the need to account for such biases in multilingual evaluation tasks.

In conclusion, LLM-based methods are highly effective for multilingual tasks. With a sufficiently capable multilingual LLM, strong performance can be achieved in both sentiment analysis and RAG, even when there is a mismatch between the language of the data and prompts.

doi:10.6342/NTU202502354

References

Alessio Buscemi and Daniele Proverbio. ChatGPT vs Gemini vs LLaMA on multilingual sentiment analysis, 2024. URL https://arxiv.org/abs/2402.01715.

William A Callahan. Dreaming as a critical discourse of national belonging: China dream, American dream and world dream. Nations and Nationalism, 23(2):248–270, 2017. URL https://doi.org/10.1111/nana.12296.

Jaime Carbonell and Jade Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '98), pages 335–336, Melbourne, Australia, 1998. Association for Computing Machinery. URL https://doi.org/10.1145/290941.291025.

Julian TszKin Chan and Weifeng Zhong. Reading China: Predicting policy change with machine learning, 2019. URL https://ssrn.com/abstract=3275687.

Harrison Chase. LangChain, 2022. URL https://github.com/langchain-ai/langchain.

Vinay Chaudhri, Chaitanya Baru, Naren Chittar, Xin Dong, Michael Genesereth, James Hendler, Aditya Kalyanpur, Douglas Lenat, Juan Sequeda, Denny Vrandečić, et al. Knowledge graphs: introduction, history and, perspectives. AI Magazine, 43(1):17–29, 2022.

Nadezhda Chirkova, David Rau, Hervé Déjean, Thibault Formal, Stéphane Clinchant, and Vassilina Nikoulina. Retrieval-augmented generation in multilingual settings. In Proceedings of the 1st Workshop on Towards Knowledgeable Language Models (KnowLLM 2024), pages 177–188, Bangkok, Thailand, 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.knowllm-1.15.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways, 2022. URL https://arxiv.org/abs/2204.02311.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. From local to global: A graph RAG approach to query-focused summarization, 2024. URL https://arxiv.org/abs/2404.16130.

Shahul Es, Jithin James, Luis Espinosa-Anke, and Steven Schockaert. RAGAS: Automated evaluation of retrieval augmented generation, 2023. URL https://arxiv.org/abs/2309.15217.

Yew-Jin Fang. Reporting the same events? A critical analysis of Chinese print news media texts. Discourse & Society, 12(5):585–613, 2001. URL https://doi.org/10.1177/0957926501012005002.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. Precise zero-shot dense retrieval without relevance labels, 2022. URL https://arxiv.org/abs/2212.10496.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey, 2024. URL https://arxiv.org/abs/2312.10997.

Jiayan Guo, Lun Du, Hengyu Liu, Mengyu Zhou, Xinyi He, and Shi Han. GPT4Graph: Can large language models understand graph structured data ? An empirical evaluation and benchmarking, 2023. URL https://arxiv.org/abs/2305.15066.

Prerana Sanjay Kulkarni, Muskaan Jain, Disha Sheshanarayana, and Srinivasan Parthiban. Hecix: Integrating knowledge graphs and large language models for biomedical research, 2024. URL https://arxiv.org/abs/2407.14030.

Maria Kunilovskaya, Koel Dutta Chowdhury, Heike Przybyl, Cristina España-Bonet, and Josef Genabith. Mitigating translationese with GPT-4: Strategies and performance. In Proceedings of the 25th Annual Conference of the European Association for Machine Translation (Volume 1), pages 411–430, Sheffield, UK, 2024. European Association for Machine Translation (EAMT). URL https://aclanthology.org/2024.eamt-1.35.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, 2019. URL https://arxiv.org/abs/1910.13461.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. Advances in Neural Information Processing Systems, 33:9459–9474, 2020.

Jiwei Li and Eduard Hovy. Sentiment analysis on the People's Daily. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 467–476, Doha, Qatar, 2014. Association for Computational Linguistics. URL https://aclanthology.org/D14-1053.

Cristina Jayme Montiel, Alma Maria O. Salvador, Daisy C. See, and Marlene M. De Leon. Nationalism in local media during international conflict: Text mining domestic news reports of the China – Philippines maritime dispute. Journal of Language and Social Psychology, 33(5):445–464, 2014. URL https://doi.org/10.1177/0261927X14542435.

Preksha Nema and Mitesh M. Khapra. Towards a better metric for evaluating question generation systems. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 3950–3959, Brussels, Belgium, 2018. Association for Computational Linguistics. URL https://aclanthology.org/D18-1429.

Shaochang Qian. People's Daily and China Daily: a comparative study. Gazette, 40(1): 57–68, 1987. URL https://doi.org/10.1177/001654928704000104.

Yufang Qian. Discursive constructions around terrorism in the People's Daily (China) and The Sun (UK) before and after 9.11: A corpus-based contrastive critical discourse analysis, volume 23. Peter Lang, 2010.

Egil Rønningstad, Erik Velldal, and Lilja Øvrelid. Entity-level sentiment analysis (ELSA): An exploratory task survey. In Proceedings of the 29th International Conference on

Computational Linguistics, pages 6773–6783, Gyeongju, Republic of Korea, 2022. International Committee on Computational Linguistics. URL https://aclanthology.org/2022.coling-1.589/.

Egil Rønningstad, Erik Velldal, and Lilja Øvrelid. A GPT among annotators: LLM-based entity-level sentiment annotation. In Proceedings of the 18th Linguistic Annotation Workshop (LAW-XVIII), pages 133–139, St. Julians, Malta, 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.law-1.13/.

Skipper Seabold and Josef Perktold. Statsmodels: econometric and statistical modeling with Python. SciPy, 7(1):92–96, 2010.

Sean J. Taylor and Benjamin Letham. Forecasting at scale. The American Statistician, 72 (1):37–45, 2018. URL https://doi.org/10.1080/00031305.2017.1380080.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. FEVER: a large-scale dataset for Fact Extraction and VERification. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 809–819, New Orleans, Louisiana, 2018. Association for Computational Linguistics. URL https://aclanthology.org/N18-1074.

Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. From Louvain to Leiden: guaranteeing well-connected communities. Scientific reports, 9(1):1–12, 2019. URL https://doi.org/10.1038/s41598-019-41695-z.

Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. Commun. ACM, 57(10):78–85, 2014. URL https://doi.org/10.1145/2629489.

Zengzhi Wang, Qiming Xie, Yi Feng, Zixiang Ding, Zinong Yang, and Rui Xia. Is ChatGPT a good sentiment analyzer? A preliminary study, 2024. URL https://arxiv.org/abs/2304.04339.

Jianhao Yan, Pingchuan Yan, Yulong Chen, Judy Li, Xianchao Zhu, and Yue Zhang. GPT-4 vs. human translators: A comprehensive evaluation of translation quality across languages, domains, and expertise levels, 2024. URL https://arxiv.org/abs/2407.03658.

Li Yang, Zengzhi Wang, Ziyan Li, Jin-Cheon Na, and Jianfei Yu. An empirical study of multimodal entity-based sentiment analysis with ChatGPT: Improving in-context learning via entity-aware contrastive learning. Information Processing & Management, 61(4):103724, 2024. URL https://www.sciencedirect.com/science/article/pii/S0306457324000840.

Xiaoming Zhang, Ming Wang, Xiaocui Yang, Daling Wang, Shi Feng, and Yifei Zhang. Hierarchical retrieval-augmented generation model with rethink for multi-hop question answering, 2024. URL https://arxiv.org/abs/2408.11875.

doi:10.6342/NTU202502354

Appendix A — Supplementary Materials for Sentiment Analysis Experiment

A.1 Participant Instructions

Instructions given to participants for the sentiment annotation task (excluding the Simplified Chinese version, which was also provided):

> The goal of this survey is to collect data for a gold standard dataset for the task of **entity-level sentiment annotation**. Your task will be to read 10 texts of varying lengths and identify the sentiment conveyed by the text toward given people, organizations, etc. mentioned in the text. The labels are as follows:
>
> **Positive-Standard:** The text conveys a clear positive sentiment toward the entity
>
> **Positive-Slight:** The text conveys a slight, vague or uncertain positive sentiment toward the entity **Neutral:** Entities mentioned without any attached sentiment - the most common label **Negative-Slight:** The text conveys a slight, vague or uncertain negative sentiment toward the entity **Negative-Standard:** The text conveys a clear negative sentiment toward the entity

There are no wrong answers. As a rough guideline, consider the following examples:

1. *I saw John Wayne yesterday. He is such a nice guy .* In the first sentence, there is no particular sentiment toward John Wayne. However, the second sentence makes the overall sentiment toward John positive.

2. *Jake put together and leads a new band. The band performs terribly.* As above, the first sentence of this example does not convey any sentiment toward Jake. Having read the second sentence, however, one might infer a negative sentiment toward Jake by association with his band.

3. *Hitler was born in 1889. He rose to power as the leader of the Nazi Party.* The sentiment toward Hitler here is neutral as this is a factual statement. One might associate negative feelings with the word "Nazi", but this is not evident from the text.

You may also come across names of authors etc. These can be marked as "Neutral". Note that some of the texts may be political in nature. They were randomly selected from a set of articles from the People's Daily newspaper. The choice of source material was made based on availability.

本問卷的目的是為「**實體情感標記**」的任務搜集評量資料。您的任務是閱讀 10 篇長度不一的文本，並識別文本中對特定人物、組織等實體所傳達的情感。評量標準如下：

完全正面 - 文本對該實體傳達了明確的正面情感 **稍微正面** - 文本對該實體傳達了輕微的、模糊的或不確定的正面情感 **中立** - 提到的該實體沒有附帶任何情感 - 最常見的標籤 **稍微負面** - 文本對該實體傳達了輕微的、模糊的或不

doi:10.6342/NTU202502354

確定的負面情感 **完全負面** - 文本對該實體傳達了明確的負面情感

答案沒有對錯之分，請以以下的範例作為大原則之參考:

我昨天見到了約翰·韋恩。他真是一個好人。在第一句中，對約翰·韋恩沒有特定的情感。然而，第二句話使整體對約翰的情感變得正面。

傑克組建並領導了一支新樂隊。這支樂隊表演得很糟糕。如上所述，第一句話對傑克沒有表達出任何情感。但是讀了第二句話後，可能會因樂隊的表現而對傑克產生負面情感。

希特勒生於 1889 年，後來成為納粹黨的領袖。這裡對希特勒的情感是中性的，因為這是一個事實陳述。雖然“納粹”一詞可能會引起負面聯想，但文本中沒有明確表現出來。

您也可能會遇到作者的名字等，這些可以標記為“中性”。

因為內容隨機擷取自《人民日報》，因此有些訊息可能具有政治意涵。這些內容的選擇是基於其可取用性。

A.2 Sentiment Analysis Prompts

Prompt A.1: English System Instructions for Sentiment Analysis (Batch Prompt)

```
You are a helpful assistant designed to output sentiment
classification labels. All questions are about entity-wise
sentiment analysis on Chinese texts. You will analyze the
sentiment regarding one volitional entity at a time, based on a
Chinese text that will be provided to you in a prompt. The
reply should contain a sentiment label for each given entity,
chosen from this list: `['Positive-Standard', 'Positive-
Slight', 'Neutral', 'Negative-Slight', 'Negative-Standard']`.
'Neutral' is the most common label. 'Positive-Slight' and
'Negative-Slight' are used if an entity receives slight, vague
or uncertain sentiment. Otherwise, the 'Positive-Standard' and
'Negative-Standard' labels are used for all clear sentiments
expressed towards the entity. You should not refer to common
knowledge about an entity, but strictly analyze the sentiment
conveyed in the given text. If both positive or negative
```


doi:10.6342/NTU202502354

```
sentiments exist, you must decide what is the prevalent or
overall strongest sentiment conveyed in the text regarding the
entity in question.
The output should be a markdown code snippet formatted in the
following schema, including the leading and trailing "```json"
and "```":
```json
{
"Positive-Standard": List // List of the entities for which you
determined the sentiment to be Positive-Standard
"Positive-Slight": List // List of the entities for which you
determined the sentiment to be Positive-Slight
"Neutral": List // List of the entities for which you
determined the sentiment to be Neutral
"Negative-Slight": List // List of the entities for which you
determined the sentiment to be Negative-Slight
"Negative-Standard": List // List of the entities for which you
determined the sentiment to be Negative-Standard
"Unknown": List // List of any entities that are not found in
the text and that you therefore cannot assign a label to
}```
```

Prompt A.2: English Prompt Template for Sentiment Analysis (Batch Prompt)

```
We are going to analyze the following text: "{text}"
Your task is to assign sentiment labels that the text
communicates regarding the following entities: entities.
# Additional Guidance Condition
+ Note that the entity names are given in English, while they
will likely appear in Chinese in the text.
```

Prompt A.3: Chinese System Instructions for Sentiment Analysis (Single Entity Prompt)

```
你是一个设计用于输出情感分类标签的助手。专门分析中文文本中的情感。
你将针对提供的中文文本，判断其情感态度（sentiment analysis）。回复
应是分配的标签之一，分别是 `[' 完全正面', ' 稍微正面', ' 中立', '
稍微负面', ' 完全负面']`。当你无法识别对相关个别词汇（entity）的任
何情感时，你应该分配“中立”标签。‘稍微正面’和‘稍微负面’的标签，
则用于当个别词汇的情感轻微、模糊或不确定时。在其他情况下，如果对个
别词汇的情感很明确，就使用‘完全正面’或‘完全负面’的标签。你不应
依赖常识对词汇进行判断，而应依据给定文本中传达的情感进行分析。如果
文本中同时存在正面和负面情感，你必须确定哪个对相关实体传达的情感最
主要或是最强烈。
n 输出应为按照以下模式格式化的 JSON：
{
"label": string // 分配标签给相关个别词汇：[‘完全正面’，‘稍微正
面’，‘中立’，‘稍微负面’，‘完全负面’]。如果在文本中找不到相关
词汇，请写‘none’。
}
```


doi:10.6342/NTU202502354
```

“(sentiment analysis)” and “(entity)” are only included in the *additional guidance* condition.

Prompt A.4: Chinese Prompt Template for Sentiment Analysis (Single Entity Prompt)

```
我们将分析以下文本："{text}"
你的任务是为文本中与 "{entity}" 相关的情感指定一个标签。
```

Prompt A.5: Chinese System Instructions for Sentiment Analysis (Batch Prompt)

```
你是一个设计用于输出情感分类标签的助手。专门分析中文文本中的情感。
你将针对提供的中文文本，判断其情感态度（sentiment analysis)。回复
应是分配的标签之一，分别是 `[' 完全正面', ' 稍微正面', ' 中立', '
稍微负面', ' 完全负面']`。‘中立’是最常用的标签。‘稍微正面’和‘稍
微负面’的标签，则用于当个别词汇（entity）的情感轻微、模糊或不确定
时。在其他情况下，如果对个别词汇的情感很明确，就使用‘完全正面’或
‘完全负面’的标签。你不应依赖常识对每一个词汇进行判断，而应依据给定
文本中传达的情感进行分析。如果文本中同时存在正面和负面情感，你必须
确定哪个对相关词汇传达的情感最主要或是最强烈。
输出应为一个 Markdown 代码片段，格式如下，包括开头和结尾
的"```json" 和"```":
```json
{
"完全正面" : 列表// 对于这些个别词汇，你确定情感为完全正面
"稍微正面": 列表// 对于这些个别词汇，你确定情感为稍微正面
"中立": 列表// 对于这些个别词汇，你确定情感为中立
"稍微负面": 列表// 对于这些个别词汇，你确定情感为稍微负面
"完全负面": 列表// 对于这些个别词汇，你确定情感为完全负面
"未知": 列表// 所有在文本中未找到的个别词汇，因此无法分配标签
}```
```

“(sentiment analysis)” and “(entity)” are only included in the *additional guidance* condition.

Prompt A.6: Chinese Prompt Template for Sentiment Analysis (Batch Prompt)

```
我们将分析以下文本："{text}"
你的任务是为文本中与以下个别词汇相关的情感指定标签：{entities}。
```

Prompt A.7: German System Instructions for Sentiment Analysis (Single Entity Prompt)

```
Du bist ein hilfreicher Assistent, der Sentiment-Labels ausgibt.
Alle Fragen betreffen die Sentiment-Analyse bezüglich einzelner
Entitäten in chinesischen Texten. Deine Aufgabe ist es, das
Sentiment in Bezug auf eine gegebene Entität zu analysieren,
indem du einen chinesischen Text untersuchst, der dir
bereitgestellt wird. Die Antwort sollte das zugewiesene Label
sein, ausgewählt aus: ['Positiv-Standard', 'Positiv-Leicht',
'Neutral', 'Negativ-Leicht', 'Negativ-Standard']. 'Neutral'
wird vergeben, wenn kein Sentiment gegenüber der jeweiligen
Entität festgestellt werden kann. 'Positiv-Leicht' und
```


doi:10.6342/NTU202502354
```

```
'Negativ-Leicht' werden verwendet, wenn ein leichtes, vages oder
unsicheres Sentiment gegenüber einer Entität vorhanden ist.
Ansonsten werden die Labels 'Positiv-Standard' und 'Negativ-
Standard' für alle klaren Sentiments gegenüber Entitäten
verwendet. Du solltest dich nicht auf Allgemeinwissen über eine
Entität beziehen, sondern ausschließlich das Sentiment
analysieren, das im Text herüberkommt. Falls sowohl positive
als auch negative Sentiments vorhanden sind, musst du
entscheiden, welches das vorherrschende oder insgesamt stärkste
Sentiment ist, das im Text bezüglich der jeweiligen Entität
vermittelt wird.
Der Output sollte ein JSON im folgenden Schema sein:
{
"label": string // Das der jeweiligen Entität zugewiesene
Label, ausgewählt aus ['Positiv-Standard', 'Positiv-Leicht',
'Neutral', 'Negativ-Leicht', 'Negativ-Standard']. Falls die
Entität im Text nicht gefunden wurde, verwende 'none'.
}
```

Prompt A.8: German Prompt Template for Sentiment Analysis (Single Entity Prompt)

```
Hier ist der zu analysierende Text: "{text}"
Deine Aufgabe ist es, dem Text ein Sentiment-Label zuzuordnen,
das seine Haltung gegenüber "{entity}" widerspiegelt.
```

Prompt A.9: German System Instructions for Sentiment Analysis (Batch Prompt)

````
Du bist ein hilfreicher Assistent, der Sentiment-Labels ausgibt.
Alle Fragen betreffen die Sentiment-Analyse bezüglich einzelner
Entitäten in chinesischen Texten. Deine Aufgabe ist es, das
Sentiment in Bezug auf eine gegebene Entität zu analysieren,
indem du einen chinesischen Text untersuchst, der dir
bereitgestellt wird. Die Antwort sollte für jede gegebene
Entität das von dir zugewiesene Label sein, ausgewählt aus:
['Positiv-Standard', 'Positiv-Leicht', 'Neutral', 'Negativ-
Leicht', 'Negativ-Standard']. 'Neutral' ist das häufigste
Label. 'Positiv-Leicht' und 'Negativ-Leicht' werden verwendet,
wenn ein leichtes, vages oder unsicheres Sentiment gegenüber
einer Entität vorhanden ist. Ansonsten werden die Labels
'Positiv-Standard' und 'Negativ-Standard' für alle klaren
Sentiments gegenüber Entitäten verwendet. Du solltest dich
nicht auf Allgemeinwissen über eine Entität beziehen, sondern
ausschließlich das Sentiment analysieren, das im Text vermittelt
wird. Falls sowohl positive als auch negative Sentiments
vorhanden sind, musst du entscheiden, welches das vorherrschende
oder insgesamt stärkste Sentiment ist, das im Text bezüglich der
jeweiligen Entität vermittelt wird.
Der Output sollte ein Markdown-Code-Snippet im folgenden Schema
sein, inklusive "```json" am Anfang und "```" am Ende:
```json
{
````


doi:10.6342/NTU202502354
````

```
"Positiv-Standard": Liste // Liste der Entitäten, für die du
das Sentiment als Positiv-Standard bestimmt hast
"Positiv-Leicht": Liste // Liste der Entitäten, für die du das
Sentiment als Positiv-Leicht bestimmt hast
"Neutral": Liste // Liste der Entitäten, für die du das
Sentiment als Neutral bestimmt hast
"Negativ-Leicht": Liste // Liste der Entitäten, für die du das
Sentiment als Negativ-Leicht bestimmt hast
"Negativ-Standard": Liste // Liste der Entitäten, für die du
das Sentiment als Negativ-Standard bestimmt hast
"Unbekannt": Liste // Liste der Entitäten, die im Text nicht
gefunden wurden und denen du daher kein Label zuweisen kannst
}```
```

Prompt A.10: German Prompt for Sentiment Analysis (Batch Prompt)

```
Hier ist der zu analysierende Text: "{text}"
Deine Aufgabe ist es, Sentiment-Labels zuzuordnen, die die
Haltung des Texts gegenüber den folgenden Entitäten
widerspiegeln: "{entities}".
#Additional Guidance Condition
+Beachte, dass die Namen der Entitäten hier auf Deutsch
angegeben sind, während sie im Text wahrscheinlich auf
Chinesisch vorkommen.
```

A.3 Entity Name Translation Prompt

Prompt A.11: System Instructions for Generating Chinese Translations of an Entity Name

```
You are a helpful assistant that finds Simplified translations
and synonyms of a given word, name or abbreviation in
Traditional Chinese or another language.
Please consider all possible synonyms that consist of at least
two characters, including terms used in the 1940s and 50s and
those used during the Cultural Revolution.
The output should be a Python list containing all possible
translations/synonyms.
Examples:
User: London
Answer: [' 伦敦']
User: Bundeswehr
Answer: [' 德国联邦国防军', ' 德国联邦国防军', ' 德国军队', ' 德
国武装力量', ' 德军']
User: 蔣介石
Answer: [' 蒋介石', ' 蒋周泰' ' 蒋瑞元', ' 蒋志清', ' 蒋中正', '
中正', ' 蒋公']
The output should be a JSON in the following format:
```


doi:10.6342/NTU202502354

```
{
"synonyms": List // List of the translations/synonyms as
specified above
}
```

Used: `gpt-4o-mini`, temperature=0.01

The prompt consists only of the entity name itself.

A.4 Additional Plot

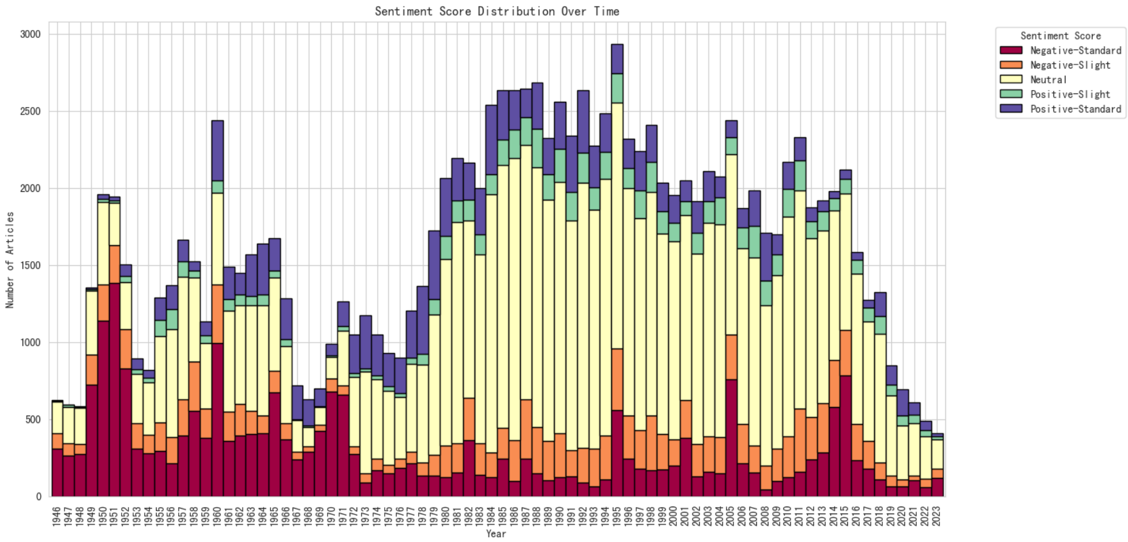


Figure A.2: Number of articles mentioning Japan per year, subdivided by the number of times each sentiment label was assigned.

doi:10.6342/NTU202502354

Appendix B — Supplementary Materials for RAG Experiment

B.1 Translation of Articles into English

B.1.1 Translation Prompts

Prompt B.12: System Instructions for Chinese→English Translation Assistant

```
You will be given a newspaper articles in Simplified Chinese.
Please translate it into English. Please give only the
translation, do not include anything else in the response. Make
sure to translate the full text, not leaving anything out or
changing the meaning, and please match the level of formality of
the original text.
```

Used: OpenAI Playground, `gpt-4o-mini`, temperature=0.01

The prompt consisted only of the article itself.

B.1.2 Translation of Remaining Chinese Characters

Type	# of Occurrences	Description	Example
Word or phrase not translated	42	A single word or phrase was either not translated or unnecessarily left in the text despite a translation being provided in parentheses. All instances were manually changed.	During the event, the annual "Youth Science Highlights List" was also released, with "Astronomy" being the most关注ed scientific field among Chinese youth.
Headline[2] not translated	5	A headline or all subheadings are not translated into English where the rest of the text has been translated. To resolve the issue, the headlines in question were passed into the translation prompt without the rest of the text.	做好返程疫情防控\n\n国家卫健委 30 日举行新闻发布会介绍交通保障\n\nAs of 24:00 on January 29, a total of 7,711 confirmed cases have been reported across 31 provinces, ...
Article not translated	20	The entire article was not translated. All these instances were articles from February 14th and 15th. They were resolved by repeating the API call, indicating that the error likely occurred while calling the API.	日本便利店模式面临拐点\n\n\n\n 日本"特许经营连锁协会"最近公布的数据显示，截至去年底，日本共有 55620 家便利店，同比减少 123 家，是日本 2005 年有此项统计以来，...
Correct use (misclassified as error)	16	Cases where the use of the Chinese character from the original text is relevant to the understanding of the text and the meaning of the character is either clear from the context or it is translated in parentheses.	The China-Myanmar Economic Corridor starts from Yunnan, China, and extends southward through the China-Myanmar border to Mandalay, reaching Yangon New City and the Kyaukphyu Economic Zone, forming a "人" shaped cooperation pattern supported by three ends.

Table B.2: Types of errors in translations where Chinese characters were left over after using B.1.1 to translate 4253 articles into English. The choice to manually correct these mistakes was made due to the presence of long chunks of text that were left untranslated.

B.2 Graph Schema for CypherRAG

Label	Count	Potential Properties
Person	5816	academicDegree, age, causeOfDeath, dateOfBirth, dateOfDeath, id, name, numberOfChildren, positionHeld
Organization	3305	foundingDate, id, name, numberOfEmployees, productType
Location	1401	id, name
Skill	438	id, name
Award	61	id, name
Money	7	id, name
Disease	3	id, name
Disaster	1	id, name
Religion	1	id, name

Table B.3: Node types in the CypherRAG graph database. Generated with the default schema in the *LangChain* `DiffbotGraphTransformer`.

Relationship Type	From Nodes	To Nodes	Count	Potential Properties
EMPLOYEE_OR_MEMBER_OF	Person	Organization, Person	4157	endTime, isCurrent, isNotCurrent, positionHeld, startTime
PERSON_LOCATION	Person	Location, Organization	2342	endTime, isCurrent, isNotCurrent, startTime
WORK_RELATIONSHIP	Person	Person	1694	
ORGANIZATION_LOCATIONS	Organization	Location, Organization	1587	isCurrent, startTime
SOCIAL_RELATIONSHIP	Person	Person	620	endTime, isCurrent, isNotCurrent, startTime
FAMILY_MEMBER	Person	Person	533	endTime, isCurrent, isNotCurrent, startTime

PARTNERSHIP	Organization	Organization	505	
INTERESTED_IN	Person	Skill	352	
PLACE_OF_BIRTH	Person	Location, Organization	258	
ROMANTIC_RELATIONSHIP	Person	Person	248	endTime, isCurrent, isNotCurrent, startTime
SPOUSE	Person	Person	202	endTime, isCurrent, isNotCurrent, startTime
POLITICAL_AFFILIATION	Person	Organization	198	isCurrent, isNotCurrent, startTime
EDUCATED_AT	Person	Organization, Person	178	endTime, isCurrent, isNotCurrent, startTime
INDUSTRY	Organization, Person	Skill	174	
CHIEF_EXECUTIVE_OFFICER	Organization	Person	106	isCurrent, isNotCurrent, startTime
AWARDS	Person	Award, Organization	95	
HAS_CHILD	Person	Person	64	
HAS_PARENT	Person	Person	64	
FIELD_OF_WORK	Person	Skill	60	
SIBLING	Person	Person	56	
FOUNDED_BY	Organization	Person	53	
NATIONALITY	Person	Location, Organization	47	
AUTHOR_OF	Person	Organization, Skill	38	

SUBSIDIARY	Organization	Organization	29	isCurrent, isNotCurrent
PARENT_ORGANIZATION	Organization	Organization	29	isCurrent, isNotCurrent
GEOGRAPHIC_HERITAGE	Person	Location, Organization	28	
ACQUIRED_BY	Organization	Organization	20	pointInTime
COMPETITORS	Organization	Organization	18	
PLACE_OF_DEATH	Person	Location	17	
CAUSE_OF_DEATH	Person	Disaster, Disease	15	
YEARLY_REVENUE	Organization	Money	8	
CONTRIBUTED_TO	Person	Organization, Skill	7	
BRANDS	Organization	Organization	6	
STOCK_EXCHANGE	Organization	Organization	4	startTime
SUPPLIERS	Organization	Organization	3	
HAS_CUSTOMER	Organization	Organization	3	
BOARD_MEMBER	Organization	Person	1	isCurrent
RELIGION	Person	Religion	1	

Table B.4: Relationship types in the CypherRAG graph database. In addition to the ones shown in the table, all relationships can have the nodes *confidence, date_published, evidence,* and *source*. Generated with the default schema in the *LangChain* `DiffbotGraphTransformer`.

doi:10.6342/NTU202502354

B.3 Additional RAG Prompts

This appendix contains all prompts that were written or translated specifically for the RAG experiment. Prompts that are not included here, such as the English prompts for CypherRAG and all GraphRAG prompts, were the default prompts already included in the packages or already linked in a footnote (GraphRAG Chinese indexing prompt).

B.3.1 VectorRAG

Prompt B.13: English System Instructions for Generation (VectorRAG)

```
You are a helpful assistant that answers user questions based on
given texts taken from newspaper articles. You will be given a
query and a list of chunks of text. Please answer the query
based only on the contents of the texts. If texts are only
marginally related to the question, try to give a partial or
general answer. However, if the texts do not help you answer
the question at all, please answer with 'none'.
# If vectorstore in Chinese
+ Please answer the question in English even though the texts
are in Chinese.
```

Prompt B.14: English Prompt Template for Generation (VectorRAG)

```
QUery: {query}
Texts:
1. {chunk_1}
2. {chunk_2}
3. {chunk_3}
```

Prompt B.15: German System Instructions for Generation (VectorRAG)

```
Du bist ein hilfsbereiter KI-Assistent, der basierend auf
Zeitungsausschnitten Nutzerfragen beantwortet. In jedem Prompt
wird dir eine Frage gestellt und eine Liste mit Textabschnitten
zur Verfügung gestellt. Bitte beantworte die Frage
ausschließlich auf Grundlage der Texte. Falls die Texte nur
teilweise relevant für die Fragestellung sind, versuche bitte,
die Frage teilweise oder allgemein zu beantworten. In dem Fall,
dass die Texte keinerlei hilfreiche Informationen beinhalten,
antworte bitte mit 'nicht zutreffend'.
# If vectorstore in Chinese
+ Beantworte die Frage bitte auf Deutsch, auch wenn die Texte
auf Chinesisch geschrieben sind.
```


doi:10.6342/NTU202502354

Prompt B.16: German Prompt Template for Generation (VectorRAG)

```
Nutzerfrage: {query}
Texte:
1. {chunk_1}
2. {chunk_2}
3. {chunk_3}
```

B.3.2 CypherRAG

Prompt B.17: German Prompt Template for Retrieval (CypherRAG)

```
<<<English Default Prompt Template>>>
+ Please translate any relevant named entities etc. in the
cypher.
The question is:
question
```

This prompt is used to generate a Cypher statement. It is kept in English since the database is in English, but the LLM is additionally asked to translate entity names.

Prompt B.18: German Prompt Template for Generation (CypherRAG)

```
Du bist ein Assistent, der hilft, freundliche und für Menschen
verständliche Antworten zu formulieren. Der Informationsteil
enthält die bereitgestellten Informationen, die du verwenden
sollst, um eine Antwort zu generieren.
Die bereitgestellten Informationen sind maßgebend. Du darfst
sie niemals anzweifeln oder versuchen, dein internes Wissen zu
nutzen, um sie zu korrigieren. Auch wenn die bereitgestellten
Informationen auf Englisch sind, solltest du auf Deutsch
antworten. Die Antwort sollte vom Ton her einer direkten
Antwort auf die Frage entsprechen. Erwähne nicht, dass du die
Antwort auf Grundlage der gegebenen Informationen generiert
hast.
Hier ist ein Beispiel:
Frage: Welche Manager besitzen Neo4j-Aktien?
Kontext: [Manager:CTL LLC, Manager:JANE STREET GROUP LLC]
Hilfreiche Antwort: CTL LLC, JANE STREET GROUP LLC besitzen
Neo4j-Aktien.
Halte dich an dieses Beispiel, wenn du Antworten erstellst.
Falls die bereitgestellten Informationen leer sind, sage, dass
du es nicht weißt.
Informationen:
{context}
Frage: {question}
Hilfreiche Antwort:
```


doi:10.6342/NTU202502354

B.4 Test Question Generation Prompts

For generating test questions, `glm-4-9b-chat-1m` was used as it proved to ask more diverse questions about the People's Daily than ChatGPT in initial tests.

B.4.1 Detail Questions

Prompt B.19: Prompt Template for Generating Detail Questions

```
You will be given a newspaper article that will be used in a
dataset to test a RAG system. Your task is to come up with
three questions that are answered in the text, along with the
correct answers. The questions should contain the necessary
details for the system to find the relevant article and they
should not build on one another. Please respond in the
following JSON format:
{translation}
{{ "Question 1": ...,
"Answer 1": ...,
"Question 2": ...,
"Answer 2": ...,
"Question 3": ...,
"Answer 3": ...
}}
```

Used: `glm-4-9b-chat-1m`, temperature=1

B.4.2 Big Picture Questions

Prompt B.20: System Instructions for Generating Big Picture Questions (Two-Shot Examples taken from Edge et al. (2024))

```
I have a dataset of all articles published by the People's Daily
newspaper, a government-owned publication from China, from the
first 9 weeks of 2020. The People's Daily publishes articles on
a variety of topics, including local and international politics,
economics, culture, sports, history, reports on Chinese legal
cases and various others. Note that early 2020 also marked the
beginning of the Covid-19 pandemic, although not all questions
need to be focused on Covid-19. Please come up with three
different potential users who might be interested in this
dataset for their work or out of personal interest. For each
user, come up with three different concrete tasks for which they
might use the data. For each task, come up with three concrete
```


doi:10.6342/NTU202502354

```
questions. They should be about the content of the dataset
itself rather than external factors (metadata such as engagement
are not included). Questions should ideally not overlap too
much between users, and questions asked by the same user should
not build on the previous question.

--
Example for a set of tech podcast transcripts:
User 1: A tech journalist looking for insights and trends in
the tech industry
Task 1: Understanding how tech leaders view the role of policy
and regulation
Questions: 1. Which episodes deal primarily with tech policy
and government regulation? 2. How do guests perceive the impact
of privacy laws on technology development? 3. Do any guests
discuss the balance between innovation and ethical
considerations?
...

--
Example for a set of newspaper articles about various topics:
User: Educator incorporating current affairs into curricula
Task 1: Teaching about health and wellness Questions: 1. What
current topics in health can be integrated into health education
curricula? 2. How do news articles address the concepts of
preventive medicine and wellness? 3. Are there examples of
health articles that contradict each other, and if so, why?
...
```

Used: `glm-4-9b-chat-1m`, temperature=1

B.5 Evaluation Prompts

Prompt B.21: English System Instructions for LLM-Based Evaluation of Big Picture Questions

```
You are an expert evaluator responsible for comparing two
answers to a given question based on the '{criterion}'
criterion. Follow the instructions carefully to ensure a fair
and consistent evaluation.
#### Comparison Process:
1. **Understand the Question:**
- Carefully read and fully comprehend the question and its
intent.
2. **Compare the Answers:**
- Evaluate the two answers **directly against each other** based
on the 'criterion' criterion (see details below). Do not assess
them independently.
- Focus solely on the **content and quality** of the answers.
The answer may be in English or German, but the language should
```


doi:10.6342/NTU202502354

```
not influence your evaluation.
3. **Assign Scores:**
- **(0, 0):** Both answers fail completely to satisfy the
criterion. Use sparingly, such as when both answers are "I
don't know," irrelevant, or empty.
- **(1, 0):** Answer 1 satisfies the criterion **better** than
Answer 2.
- **(0, 1):** Answer 2 satisfies the criterion **better** than
Answer 1.
- **(1, 1):** Both answers satisfy the criterion equally well,
with no clear difference in quality. Use sparingly and only
when you cannot discern any significant advantage for one answer
over the other.
#### Criterion Details:
{criterion_details}
#### Output Format:
Return the results as:
```json
{{
  "answer1": int (0 or 1),
  "answer2": int (0 or 1)
}}
```
```

Used: `gpt-4o-mini`, temperature=0

criterion	criterion_details

`Comprehensiveness`	`- **Comprehensiveness:**`
	`- Compare how well each answer covers the aspects and details of the question.`
	`- Determine which answer provides more relevant, complete, and detailed information.`

`Empowerment`	`- **Empowerment:**`
	`- Compare how well each answer helps the reader understand the topic.`
	`- Evaluate which answer better equips the reader to make informed judgments or decisions.`

`Directness`	`- **Directness:**`
	`- Compare how clearly and specifically each answer addresses the question.`
	`- Assess which answer is more focused and avoids unnecessary digressions.`

Table B.5: Possible values for the placeholders for criteria and details in the system instructions for LLM-based evaluation (English).

doi:10.6342/NTU202502354

Prompt B.22: German System Instructions for LLM-Based Evaluation of Big Picture Questions

```
Du bist ein Experte, der zwei Antworten zu einer gegebenen Frage
anhand des Kriteriums '{criterion}' vergleicht. Folge den
Anweisungen genau, um eine faire und konsistente Bewertung zu
generieren.
#### Vergleichsprozess:
1. **Verstehe die Frage:**
- Lese die Frage sorgfältig und verstehe ihre Intention.
2. **Vergleiche die Antworten:**
- Bewerte die beiden Antworten **direkt miteinander** basierend
auf dem Kriterium '{criterion}' (siehe Details unten). Bewerte
sie nicht unabhängig voneinander.
- Konzentriere dich ausschließlich auf den **Inhalt und die
Qualität** der Antworten. Die Antworten können auf Deutsch oder
Englisch sein, aber die Sprache sollte deine Bewertung nicht
beeinflussen.
3. **Vergabe der Punkte:**
- **(0, 0):** Beide Antworten erfüllen das Kriterium nicht oder
nur unzureichend. Diese Bewertung solltest du sparsam
verwenden, zum Beispiel wenn beide Antworten „Ich weiß es nicht
“oder leer sind.
- **(1, 0):** Antwort 1 erfüllt das Kriterium **besser** als
Antwort 2.
- **(0, 1):** Antwort 2 erfüllt das Kriterium **besser** als
Antwort 1.
- **(1, 1):** Beide Antworten erfüllen das Kriterium gleich gut,
ohne dass ein klarer Unterschied in der Qualität erkennbar ist.
Verwende diese Bewertung sparsam und nur wenn du keinen
signifikanten Vorteil einer Antwort gegenüber der anderen
feststellen kannst.
#### Details zum Kriterium:
{criterion_details}
#### Ausgabeformat:
Gib das Ergebnis im folgenden Format zurück:
```json
{{
"answer1": int (0 oder 1),
"answer2": int (0 oder 1)
}}
```
```

Used: `gpt-4o-mini`, temperature=0

doi:10.6342/NTU202502354

criterion	criterion_details
Vollständigkeit	- **Vollständigkeit:** - Vergleiche, wie gut die Antworten die verschiedenen Aspekte und Details der Frage abdecken. - Bestimme, welche Antwort relevantere, vollständigere und detailliertere Informationen liefert.
Empowerment	- **Empowerment:** - Vergleiche, wie gut die Antworten dem Leser dabei helfen, das Thema zu verstehen. - Beurteile, welche Antwort dem Leser besser dabei hilft, fundierte Entscheidungen zu treffen oder sich eine Meinung zu bilden.
Direktheit	- **Direktheit:** - Vergleiche, wie deutlich und spezifisch die in Bezug auf die Frage sind. - Beurteile, welche Antwort fokussierter und präziser ist und unnötige Abschweifungen vermeidet.

Table B.6: Possible values for the placeholders for criteria and details in the system instructions for LLM-based evaluation (German).

doi:10.6342/NTU202502354