



Text Obsoleteness Detection using Large Language Models

Rishav Ranaut

rishav.ranaut82@gmail.com

Department of Computer Science and Engineering, Indian
Institute of Technology Patna
Patna, Bihar, India

Adam Jatowt

adam.jatowt@uibk.ac.at

Department of Computer Science & Digital Science Center,
University of Innsbruck
Innsbruck, Innsbruck, Austria

Sriparna Saha

sriparna.saha@gmail.com

Department of Computer Science and Engineering, Indian
Institute of Technology Patna
Patna, Bihar, India

Manish Gupta

gmanish@microsoft.com

Bing, Microsoft
Hyderabad, Telangana, India

Abstract

Maintaining accurate and up-to-date information is a persistent challenge for large-scale knowledge repositories, where outdated content can compromise their value. In this paper, we present a Multitask learning framework that uses Large Language Models (LLMs) for two tasks: semantic update detection and semantic update necessity prediction. The update detection task identifies obsoleteness by comparing older and newer text versions, while the update necessity prediction task determines whether an update is required based on a given context. To support these tasks, we curate a specialized dataset from Wikipedia called SEMUPDATES, focusing on frequently updated articles. Our experiments with five LLMs across four datasets in zero-shot, few-shot, and fine-tuned settings demonstrate that fine-tuning significantly enhances performance. In the multitask learning setup, Qwen delivers the best overall performance, while Mistral achieves the highest accuracy on individual tasks when fine-tuned separately. However, the performance differences across models are not substantial, suggesting that multiple LLMs can be effectively adapted for content update automation. These findings highlight the potential of LLMs in detecting and predicting obsolescence, providing a scalable solution for maintaining the timeliness of digital knowledge repositories.

CCS Concepts

• Information systems → Information systems applications.

Keywords

Text Obsoleteness, LLM, MultiTask learning

ACM Reference Format:

Rishav Ranaut, Sriparna Saha, Adam Jatowt, and Manish Gupta. 2025. Text Obsoleteness Detection using Large Language Models. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '25), July 13–18, 2025, Padua, Italy*. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/3726302.3730254>



This work is licensed under a Creative Commons Attribution 4.0 International License.
SIGIR '25, July 13–18, 2025, Padua, Italy
© 2025 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-1592-1/2025/07
<https://doi.org/10.1145/3726302.3730254>

1 Introduction

Maintaining the timeliness and relevance of content in large-scale knowledge repositories is a persistent challenge [21], as information can quickly become outdated in the face of rapidly evolving knowledge [6]. Outdated content undermines the accuracy and reliability of such repositories, leading to user frustration and diminished trust. Robust mechanisms are needed not only to detect obsoleteness in content but also to predict when updates are necessary, ensuring that the information repositories and, in general, information services remain current and reliable over time [23]. Similarly, having means to estimate the likelihood of information obsoleteness of content produced by Generative AI would be quite beneficial.

In this paper, we explore how Large Language Models (LLMs) can be utilized to manage and improve text timeliness estimation through two tasks: *semantic update detection* (SUD) and *semantic update necessity prediction* (SUNP). SUD involves determining whether a factual update (such as a change in dates, numerical values, scores, episodes, status, actors, etc.) has occurred by comparing an older version of content with a newer one. This can be useful when an LLM needs to compare the target statement against some evidence (e.g., related statements collected from the Web). SUNP, on the other hand, determines whether a semantic update is likely to be required for a given text. This task can be helpful, for example, for LLMs to estimate the likelihood of text obsoleteness in the absence of any evidence. The two tasks are interrelated as both require models to have a contextual understanding of text's timeliness.

Related Works Recently, there has been significant research into understanding and predicting the temporal validity duration of textual content, with applications ranging from enhancing document credibility to improving information retrieval systems [3, 15]. Additionally, the challenge of temporal commonsense reasoning has garnered attention [24], emphasizing the need for language models to grasp the typical temporal context of events and actions. This is essential for various NLP tasks such as timeline summarization and temporal inference [2, 24, 27]. Previous studies have highlighted the gaps in current models' ability to handle the proposed tasks, in particular, identifying the semantic changes happening in a text over a time period [18, 22]. In the current research, we explore these dimensions using LLMs and we focus on two key tasks: SUD and SUNP. SUD is useful for maintaining accurate and current

Table 1: Summary of Dataset Statistics (SU stands for semantic update)

Dataset	# SU samples	# No SU samples	Total # Samples
Edit-Intentions [25]	1,148	4,538	5,686
Wiki-TIDE [5]	1,571	474	2,045
One Document, Many Revisions [19]	354	0	354
SemUpdates (Ours)	1,174	4,461	5,635
<i>Total</i>	4,247	9,473	13,720
Sample size used for our tasks			
SUD Task Dataset	4,247	4,247	8494
SUNP Task Dataset	4,247	4,247	8494
MTL Task Dataset	SUD: 4,247 SUNP: 4,247	SUD: 4,247 SUNP: 4,247	16,988

knowledge bases, automating content moderation, fact-checking, and monitoring changes in legal or compliance documents. On the other hand, SUNP can be linked to Retrieval-Augmented Generation (RAG) systems [10], where predicting potential content changes can help in making more relevant and timely retrieval calls [14], enhancing the quality and relevance of generated responses. Our work aligns with and contributes to the broader understanding of temporal reasoning in LMs, providing insights into the dynamic nature of textual information [12].

To facilitate our research, we curate a specialized dataset¹ from Wikipedia, a platform where content often evolves based on factual changes in the real world, making it an appropriate source for studying text obsolescence. We select articles that are prone to frequent updates, ensuring that the dataset is optimized for both tasks (SUD and SUNP). This dataset, which we call SEMUPDATES, is used for training and evaluating the Multitask learning (MTL) framework, where a single model is required to perform both tasks. We explore two distinct approaches for Multi-Task Learning (MTL): (1) Instruction-Tuned MTL [8], where the model learns to generalize across both tasks based on task descriptions, and (2) Task-Specific Heads MTL [26], where task routing ensures specialized learning for each task. Through the experiments, we investigate knowledge transfer between SUD and SUNP, assessing whether learning from one task enhances performance on the other. Overall, we make the following contributions:

1) We systematically explore how Large Language Models (LLMs) can be leveraged for two closely related tasks: detecting when content becomes obsolete and predicting when future updates will be needed. 2) We introduce SEMUPDATES, a novel dataset, along with a scalable pipeline for identifying frequently updated content, providing a valuable resource for studying long-term semantic drift. 3) We propose and evaluate an MTL framework that incorporates both instruction tuning and task-specific heads, enabling a single model to handle both tasks concurrently while maximizing knowledge transfer.

2 Datasets

We utilized several existing datasets, as listed in Section 2.1. In addition, we also create our own dataset SEMUPDATES, described in Section 2.2. Table 1 shows the summary of basic dataset statistics.

2.1 Details of Existing Datasets

Edit-Intentions [25] contains revision IDs from various Wikipedia pages. Each revision ID is annotated based on the type of edit, such

¹Code available at <https://github.com/rishavranaud/Text-Obsolescence>

Table 2: Examples from our SEMUPDATES dataset. Updates are highlighted in blue.

Wiki-ref	Old Content	New Content	GT
Midnight Special (film)	Midnight Special has received positive reviews from critics. It holds a rating of 85% on Rotten Tomatoes, based on 20 reviews, and an average rating of 6.8 out of 10. On Metacritic, the film has a score of 78 out of 100, indicating generally favorable reviews.	Midnight Special has received positive reviews from critics, with an 86% rating on Rotten Tomatoes, based on 22 reviews and an average rating of 6.8 out of 10. On Metacritic, the film has a score of 77 out of 100, indicating generally favorable reviews.	SU=1
Nichols algebra	This root system is the smallest member of an infinite series. The images are from ref name=CL15, where this example is also discussed in detail.	This root system represents the smallest element of an infinite series. The images originate from ref name=CL15, where this example is also explained thoroughly.	SU=0
Battle of Plataea	The estimated strength of Herodotus is 300,000, with an additional 50,000 from his Greek allies. Diodorus estimates 500,000. Modern consensus ranges from 100,000 to 120,000 , including Greek allies. The casualties are estimated as follows: Ephorus and Diodorus at 10,000 , 1,360 by Plutarch, Herodotus at 759 , and modern consensus at 50,000-70,000 .	The estimated strength of Herodotus is 300,000, with an additional 50,000 from his Greek allies. Diodorus estimates around 500,000. Modern consensus ranges between 70,000 and 120,000 , including Greek allies. Casualties are estimated at 10,000 according to Ephorus and Diodorus , 1,360 by Plutarch, 159 by Herodotus. Modern consensus estimates casualties at around 10,000 to 20,000 .	SU=1

as fact update, clarification, elaboration, and others. For each revision ID, we retrieved the corresponding old and new content from the Wikipedia pages. We refer to instances labeled as “Fact Updates” as label 1 (i.e., Semantic Update=1) and others as label 0, indicating the absence of a semantic update.

Wiki-TIDE [5] contains old/new content pairs from Wikipedia. We take examples with labels 1 (described as “they may be semantically similar or different, yet, without a fundamental change”) and 2 (“they differ due to fundamental changes”). We labeled these instances initially using ChatGPT-3.5 Turbo-Instruct, followed by human annotation to validate the results, and consider the validated ones as SU=1. The remaining examples were labeled as SU=0 (i.e., no semantic update).

One Document, Many Revisions [19] dataset contains contiguous revisions of Wikipedia pages over time that identify the intentions behind edits over each revision. From this dataset, we gathered rows labeled as “Fact Updates only” as SU=1.

2.2 SEMUPDATES Dataset Curation

The quality and diversity of our dataset are critical for accurately detecting semantic updates. To construct SEMUPDATES, we carefully augment existing datasets through a multi-stage process using the pipeline illustrated in Fig. 1 that captures genuine content changes. We start by selecting Wikipedia pages from a broad range of categories—including sports, education, politics, health, tourism, employment, wildlife conservation, law, trade, and technology—to ensure wide topical coverage. For each page, we extract all available revision IDs to build a comprehensive pool of candidate revision pairs. Two complementary strategies are employed to select revision pairs: one pairs the most recent revision with the median revision (with the median serving as an earlier state), and the other pairs revisions based on a predefined time interval (δT). Both approaches are designed to increase the likelihood of capturing revisions that contain significant content changes.

To filter out pairs with only trivial modifications, we compute the symmetric difference in the number of unique words between the old and new versions. Only those pairs where this difference is more than a specific threshold (δ) are retained, ensuring that our dataset emphasizes meaningful relationship between old and new

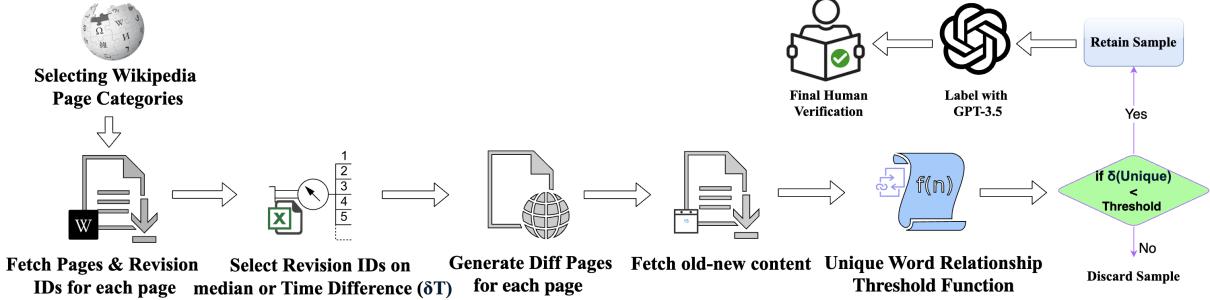


Figure 1: Dataset Generation Pipeline.

pair of sentences rather than superficial edits.

Following this quantitative filtering, we subject each revision pair to a two-stage annotation process. Initially, ChatGPT-3.5 Turbo-Instruct provides a preliminary classification. This is then refined by human annotators who focus on identifying substantial factual modifications—such as changes in numerical values, dates, scores, episode counts, and status updates. Revision pairs exhibiting only minor stylistic adjustments (e.g., punctuation changes or paraphrasing) are labeled as “No Semantic Update.” This combination of automated and manual annotation ensures that our final labels reliably distinguish between semantic updates and inconsequential revisions.

The resulting dataset, SEMUPDATES, comprises a diverse collection of examples where each pair clearly reflects either a significant update or a minor change. Table 2 presents selected examples from the dataset. Further for the SUNP task, we use the old versions of SU=1 samples across all 4 datasets as label 1. Similarly, for label 0, we use old versions of SU=0 samples collected from our pipeline. Overall, our dataset curation process is driven by the goal of capturing real-world editing behaviors, thereby providing a solid foundation for the evaluation and development of semantic update detection methods.

3 Experiments

We experimented with several state-of-the-art LLMs, including Mistral 7B [13], Qwen-2 7B [4], LLaMA-3 8B [1], Flan-T5 [7], and GPT-4 [17], using QLoRA configurations [9, 11]. The models were loaded in 4-bit precision with LoRA parameters $r = 16$, lora_alpha = 8, lora_dropout = 0.05, and target modules (q_proj, k_proj, v_proj, o_proj).

Instead of manually testing multiple prompts, we provided a raw task description to DSPY [16, 20], which optimized the final prompt automatically. This optimization was performed using Ollama with LLaMA-3.1(8B), ensuring a systematic approach. The optimized prompt was then manually checked and applied consistently across all models to maintain a fair comparison. We evaluated zero-shot and few-shot capabilities under these controlled conditions. Additionally, we explored fine-tuning by splitting the dataset into training (70%), validation (10%), and test (20%) sets to assess performance improvements.

For the SUD task, we provide both the old and new content to the LLM shown in Fig. 2, along with the prompt that describes the task and label definitions as follows. “Determine whether a text passage has been factually updated by identifying changes in date,

Table 3: Performance comparison for SUD and SUNP Tasks.

Model	Technique	SUD			SUNP				
		Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
Flan-T5	Zero-Shot	0.49	0.48	0.93	0.64	0.50	0.38	0.04	0.07
	Few-Shot	0.47	0.47	0.85	0.61	0.50	0.41	0.05	0.09
	Fine-tuned	0.80	0.82	0.78	0.80	0.72	0.68	0.84	0.75
LLaMA3	Zero-Shot	0.51	0.50	0.97	0.66	0.49	0.49	1.0	0.65
	Few-Shot	0.64	0.58	0.93	0.71	0.56	0.52	0.97	0.68
	Fine-tuned	0.78	0.76	0.82	0.79	0.78	0.84	0.70	0.76
Mistral	Zero-Shot	0.54	0.51	0.95	0.66	0.47	0.47	0.92	0.63
	Few-Shot	0.67	0.72	0.54	0.62	0.49	0.46	0.30	0.36
	Fine-tuned	0.81	0.86	0.74	0.79	0.78	0.85	0.68	0.75
Qwen2	Zero-Shot	0.60	0.55	0.93	0.69	0.56	0.54	0.73	0.62
	Few-Shot	0.64	0.59	0.85	0.70	0.51	0.50	0.90	0.64
	Fine-tuned	0.79	0.81	0.76	0.78	0.77	0.81	0.70	0.75
GPT-4	Zero-Shot	0.75	0.74	0.76	0.75	0.58	0.61	0.42	0.50
	Few-Shot	0.76	0.74	0.77	0.76	0.63	0.64	0.54	0.59

numbers, scores, statuses, or other relevant information between two given sentences. Provide a binary answer (Yes/No) indicating if the new sentence represents a factual update to the old sentence.’ Old sentence: [old] [SEP] New sentence: [new] Answer:’

The SUNP task aims to predict whether a given text will require a factual update in the future. This is particularly useful for maintaining the accuracy of evolving information sources, such as news articles, regulatory guidelines, and technical documentation. The key idea behind this task is that certain textual statements inherently carry a higher probability of change over time—such as numerical values, dates, or status indicators—while others remain static. Automating this prediction can facilitate proactive content monitoring, ensuring that information remains current and reducing the risk of outdated or misleading content.

To perform the SUNP task, we provide a sentence (old content) as input to the LLM shown in Fig. 2, which then predicts, based on its understanding of the content, whether a factual update is likely to occur. We use the following prompt: “Analyze the given text based on its contextual understanding to determine whether any factual updates (e.g., date changes, numerical updates, score modifications, or status changes) are likely to occur in the future. Return a response indicating ‘Yes’ if an update is predicted and ‘No’ otherwise. Text: [sentence] Answer:”

The model’s ability to interpret semantic and contextual nuances is crucial for this prediction. For example, the model should be able to reason that (1) Historical facts are less likely to change over time. (2) Statements related to rapidly evolving topics or domains, such as technology or politics, are more likely to be updated.

Results for Individual Tasks: Table 3 summarizes performance of LLMs on SUD and SUNP. Precision, recall, and F1 focus on the SU=1 class (semantic updates). **Zero-Shot Performance** GPT-4 achieves the highest accuracy in SUD (0.75) and SUNP (0.58), demonstrating

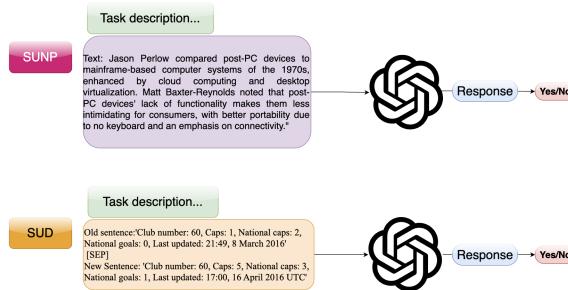


Figure 2: Example Inputs of both Tasks

strong baseline performance. Qwen2 follows with SUD (0.60) and SUNP (0.56). Flan-T5 struggles (0.49, 0.50), while LLaMA3 exhibits high recall (0.97, 1.0) but low precision, indicating overgeneralization. Mistral performs moderately (0.64, 0.52), suggesting potential for task-specific improvement.

Few-Shot Performance Providing two few-shot examples yields varying improvements in SUD and SUNP task. GPT-4 maintains strong performance in SUD (0.76) and achieves moderate gains in SUNP (0.63). Mistral improves in SUD (0.67) but struggles to generalize in SUNP (0.49). LLaMA3 benefits from the few-shot examples, increasing from 0.55 to 0.61 in SUD and 0.50 to 0.56 in SUNP. Qwen2 remains consistent, with SUD improving ($0.60 \rightarrow 0.64$) but SUNP slightly decreasing ($0.56 \rightarrow 0.51$). Flan-T5 declines in SUD ($0.49 \rightarrow 0.47$) and remains unchanged in SUNP (0.50), indicating limited adaptability to the semantic update tasks across both zero-shot and few-shot settings. On average, models improve by 5.8% in SUD and 1.8% in SUNP, underscoring the benefits of limited in-context learning for semantic update detection and prediction.

Fine-Tuned Models LoRA-based fine-tuning [11] significantly improved performance across all models. Mistral achieved the highest accuracy with 0.81 (SUD) and 0.78 (SUNP), followed closely by Qwen2 and LLaMA3. Flan-T5 also saw substantial gains, particularly in SUNP. Fine-tuning boosted accuracy by an average of 18.5% (SUD) and 23.5% (SUNP) over zero/few-shot baselines, highlighting the necessity of task-specific adaptation.

3.1 Multitask Learning

We further investigate whether LLMs can simultaneously handle SUD and SUNP tasks together, given their shared reliance on natural language understanding.

Our approach utilizes a Multi-task Learning (MTL) framework (Fig. 3) with Flan-T5 and quantized LLMs (LLaMA, Mistral, Qwen), ensuring adaptability across architectures. For decoder-based models (e.g., LLaMA, Mistral, Qwen), we apply QLoRA fine-tuning [9, 11], loading the model in 4-bit precision and adapting the query, key, value, and output projection layers (q_proj , k_proj , v_proj , o_proj). This facilitates efficient adaptation through rank-16 adapters, an alpha scaling factor of 8, and a dropout rate of 0.05, maintaining a balance between parameter efficiency and generalization. Classification is conducted using the last hidden state, which provides the most contextualized representation of the input.

For Flan-T5, an encoder-decoder model, we fine-tune only the encoder along with task-specific classification heads. Since classification does not require sequence generation, training only the encoder reduces computational cost while preserving performance.

Table 4: Performance of the LLMs in Multitask finetuning.

Model	SUD				SUNP			
	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
Flan-T5	0.79	0.81	0.76	0.78	0.77	0.78	0.76	0.77
LLaMA3	0.81	0.85	0.75	0.80	0.77	0.85	0.66	0.74
Mistral	0.81	0.86	0.75	0.80	0.77	0.84	0.67	0.75
Qwen2	0.82	0.85	0.78	0.81	0.77	0.83	0.69	0.75

Table 5: Performance of the fine-tuned LLMs in the Multitask Learning scenario with separate heads

Model	SUD				SUNP			
	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
Flan-T5	0.80	0.84	0.74	0.79	0.79	0.82	0.73	0.78
LLaMA3	0.79	0.91	0.64	0.75	0.79	0.89	0.65	0.75
Mistral	0.79	0.84	0.71	0.77	0.78	0.85	0.68	0.76
Qwen2	0.81	0.85	0.76	0.80	0.78	0.82	0.72	0.76

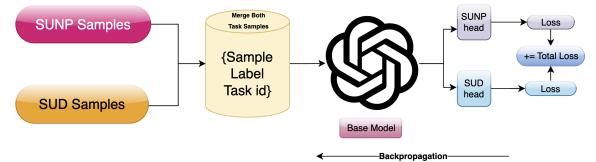


Figure 3: Task specific heads for Multitask learning with shared base model

The encoder outputs are passed through task-specific classification heads, each consisting of a LayerNorm and a linear transformation to generate logits. A task identifier (task id) routes inputs to the appropriate classification head, allowing knowledge sharing while preserving task-specific specialization. This design ensures efficient parameter utilization while maintaining strong task performance.

We also conduct experiments with standard supervised fine-tuning (SFT), where models are trained on both tasks without task-specific classification heads. Instead, we follow the conventional SFT setup, using the same LoRA configuration for decoder-based models while fully fine-tuning Flan-T5. This approach enables the model to learn both tasks within a shared representation space without explicit task separation.

Tables 4 and 5 show that both multitask fine-tuning and task-specific training with separate heads yield strong results, confirming a single model can learn multiple tasks effectively. Multitask fine-tuning offers the advantage of efficiency, reducing the need for multiple specialized models while maintaining competitive accuracy. On the other hand, using separate heads allows for more targeted learning, ensuring each task benefits from dedicated optimization. Both approaches are beneficial, depending on whether the goal is to streamline training or to fine-tune task-specific representations. Among the models, Qwen emerged as the best-performing one while all models performed competitively well.

4 Conclusion

In this work, we explored the capabilities of Large Language Models (LLMs) in detecting and predicting text obsolescence through Semantic Update Detection (SUD) and Semantic Update Necessity Prediction (SUNP). To support these tasks, we introduced SEMUPDATES, a Wikipedia-based dataset, and developed an automated pipeline for extracting and processing frequently updated content. Our experiments with five LLMs show that fine-tuning significantly improves performance, with Qwen excelling in Multitask learning and Mistral slightly outperforming on individual tasks.

References

- [1] AI@Meta: Llama 3 model card (2024), https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md
- [2] Allan, J., Gupta, R., Khandelwal, V.: Temporal summaries of new topics. In: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 10–18 (2001)
- [3] Almquist, A., Jatowt, A.: Towards content expiry date determination: predicting validity periods of sentences. In: Advances in Information Retrieval: 41st European Conference on IR Research, ECIR 2019, Cologne, Germany, April 14–18, 2019, Proceedings, Part I 41. pp. 86–101. Springer (2019)
- [4] Bai, J., Bai, S., Chu, Y., Cui, Z., Dang, K., Deng, X., Fan, Y., Ge, W., Han, Y., Huang, F., et al.: Qwen technical report. arXiv preprint arXiv:2309.16609 (2023)
- [5] Borkakoti, H., Espinosa-Anke, L.: Wikitide: A wikipedia-based timestamped definition pairs dataset (2023), <https://arxiv.org/abs/2308.03582>
- [6] Chaibet, S., Hnich, B., Mrad, A.B.: Data obsolescence detection in the light of newly acquired valid observations. Applied Intelligence **52**(14), 16532–16554 (2022)
- [7] Chung, H.W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, E., Wang, X., Dehghani, M., Brahma, S., Webson, A., Gu, S.S., Dai, Z., Suzgun, M., Chen, X., Chowdhery, A., Narang, S., Mishra, G., Yu, A., Zhao, V., Huang, Y., Dai, A., Yu, H., Petrov, S., Chi, E.H., Dean, J., Devlin, J., Roberts, A., Zhou, D., Le, Q.V., Wei, J.: Scaling instruction-finetuned language models (2022). <https://doi.org/10.48550/ARXIV.2210.11416>, <https://arxiv.org/abs/2210.11416>
- [8] Chung, H.W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, Y., Wang, X., Dehghani, M., Brahma, S., et al.: Scaling instruction-finetuned language models. Journal of Machine Learning Research **25**(70), 1–53 (2024)
- [9] Dettmers, T., Pagnoni, A., Holtzman, A., Zettlemoyer, L.: Qlora: Efficient finetuning of quantized llms. Advances in Neural Information Processing Systems **36** (2024)
- [10] Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., Wang, H.: Retrieval-augmented generation for large language models: A survey. arXiv preprint arXiv:2312.10997 (2023)
- [11] Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: Lora: Low-rank adaptation of large language models (2021), <https://arxiv.org/abs/2106.09685>
- [12] Jain, R., Sojitra, D., Acharya, A., Saha, S., Jatowt, A., Dandapat, S.: Do language models have a common sense regarding time? revisiting temporal commonsense reasoning in the era of large language models. In: Bouamor, H., Pino, J., Bali, K. (eds.) Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pp. 6750–6774. Association for Computational Linguistics, Singapore (Dec 2023). <https://doi.org/10.18653/v1/2023.emnlp-main.418>, <https://aclanthology.org/2023.emnlp-main.418>
- [13] Jiang, A.Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D.S., de las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L.R., Lachaux, M.A., Stock, P., Scao, T.L., Lavril, T., Wang, T., Lacroix, T., Sayed, W.E.: Mistral 7b (2023), <https://arxiv.org/abs/2310.06825>
- [14] Kasai, J., Sakaguchi, K., Le Bras, R., Asai, A., Yu, X., Radev, D., Smith, N.A., Choi, Y., Inui, K., et al.: Realtime qa: what's the answer right now? Advances in Neural Information Processing Systems **36** (2024)
- [15] Kawai, H., Jatowt, A., Tanaka, K., Kunieda, K., Yamada, K.: Chronoseeker: Search engine for future and past events. In: Proceedings of the 4th International Conference on Uniquitous Information Management and Communication. pp. 1–10 (2010)
- [16] Khattab, O., Singhvi, A., Maheshwari, P., Zhang, Z., Santhanam, K., Vardhaman, S., Haq, S., Sharma, A., Joshi, T.T., Moazam, H., et al.: Dspy: Compiling declarative language model calls into self-improving pipelines. arXiv preprint arXiv:2310.03714 (2023)
- [17] OpenAI, et al.: Gpt-4 technical report (2024), <https://arxiv.org/abs/2303.08774>
- [18] Periti, F., Montanelli, S.: Lexical semantic change through large language models: a survey. ACM Comput. Surv. **56**(11) (Jun 2024). <https://doi.org/10.1145/3672393>
- [19] Rajagopal, D., Zhang, X., Gamon, M., Jauhar, S.K., Yang, D., Hovy, E.: One document, many revisions: A dataset for classification and description of edit intents. In: Proceedings of the Thirteenth Language Resources and Evaluation Conference. pp. 5517–5524. European Language Resources Association, Marseille, France (Jun 2022). <https://aclanthology.org/2022.lrec-1.591>
- [20] Sahoo, P., Singh, A.K., Saha, S., Jain, V., Mondal, S., Chadha, A.: A systematic survey of prompt engineering in large language models: Techniques and applications. arXiv preprint arXiv:2402.07927 (2024)
- [21] Thomer, A.K., Starks, J.R., Rayburn, A., Lenard, M.C.: Maintaining repositories, databases, and digital collections in memory institutions: An integrative review. Proceedings of the Association for Information Science and Technology **59**(1), 310–323 (2022)
- [22] Wang, R., Choi, M.: Large language models on lexical semantic change detection: An evaluation. arXiv preprint arXiv:2312.06002 (2023)
- [23] Wang, S., Zhu, Y., Liu, H., Zheng, Z., Chen, C., Li, J.: Knowledge editing for large language models: A survey. ACM Computing Surveys **57**(3), 1–37 (2024)
- [24] Wenzel, G., Jatowt, A.: An overview of temporal commonsense reasoning and acquisition (2023), <https://arxiv.org/abs/2308.00002>
- [25] Yang, D., Halfaker, A., Kraut, R., Hovy, E.: Identifying semantic edit intentions from revisions in Wikipedia. In: Palmer, M., Hwa, R., Riedel, S. (eds.) Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 2000–2010. Association for Computational Linguistics, Copenhagen, Denmark (Sep 2017). <https://doi.org/10.18653/v1/D17-1213>, <https://aclanthology.org/D17-1213>
- [26] Zhang, Y., Yang, Q.: An overview of multi-task learning. National Science Review **5**(1), 30–43 (2018)
- [27] Zhou, B., Khashabi, D., Ning, Q., Roth, D.: “going on a vacation” takes longer than “going for a walk”: A study of temporal commonsense understanding. In: Inui, K., Jiang, J., Ng, V., Wan, X. (eds.) Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 3363–3369. Association for Computational Linguistics, Hong Kong, China (Nov 2019). <https://doi.org/10.18653/v1/D19-1332>, <https://aclanthology.org/D19-1332>