Decoupling Reasoning from Language: A Hybrid Architecture for Temporal Reasoning in Vietnamese

Xuan-Truong Quan, Duc Do Minh, Dat Nguyen-Tat, Xuan-Son Quan, Vinh Nguyen Van
University of Engineering and Technology (UET)
xuantruongvnuet@gmail.com, dominhduc@vnu.edu.vn
nguyentatdat2811@gmail.com, quanxuanson2004@gmail.com, vinhnv@vnu.edu.vn

Abstract

Temporal reasoning remains a formidable challenge for Large Language Models (LLMs), particularly in low-resource languages where annotated data is scarce. This paper addresses the Date Arithmetic task in Vietnamese by proposing a novel hybrid architecture that decouples core reasoning capabilities from language-specific representation. Our approach combines a powerful Flan-T5-base reasoning engine, fine-tuned on a vast English dataset, with a lightweight Qwen2.5-1.5B-Instruct model serving as a Cross-Lingual Semantic Adapter. This adapter first parses unstructured Vietnamese queries into a canonical English format for the Flan-T5 engine, and subsequently translates the engine's English output back into fluent Vietnamese. On a public benchmark, our method achieves state-of-the-art accuracy of 0.98, matching the performance of the most advanced proprietary LLMs like Gemini-2.5 Pro while drastically outperforming standard fine-tuning methods. This work demonstrates that by separating the reasoning module from the linguistic interface, we can effectively transfer sophisticated temporal logic from high-resource to low-resource languages, paving the way for more efficient and accessible robust cross-lingual NLP systems.

1 Introduction

Temporal reasoning, which is the ability to comprehend and manipulate time-related concepts, is a cornerstone of natural language understanding. While recent Large Language Models (LLMs) have demonstrated impressive performance on temporal tasks, their success is heavily reliant on massive, high-resource datasets, predominantly in English (Chu et al., 2024; Tan et al., 2023). This dependency creates a significant performance gap when applying these models to low-resource languages like Vietnamese.

The Date Arithmetic task, which requires precise date calculations, exemplifies this challenge. For Vietnamese, the available dataset comprises merely 3,000 samples. In stark contrast, its English counterpart contains over 400,000 examples. Directly fine-tuning a model on such a small dataset inevitably leads to overfitting and poor generalization, failing to capture the underlying logical principles of temporal arithmetic.

To surmount this data scarcity problem, we introduce a novel approach centered on a powerful principle: decoupling reasoning from language. Instead of training a single monolithic model to handle both Vietnamese understanding and temporal calculation, we separate these concerns. Our contributions are threefold:

1. We propose a hybrid architecture that combines a specialized English temporal reasoning engine with a versatile Vietnamese language adapter.

2. We employ a lightweight Qwen2.5-1.5B-Instruct model as a Cross-Lingual Semantic Adapter, using prompt engineering to bidirectionally map between natural Vietnamese queries and a canonical English format.

3. We empirically demonstrate that our decoupled approach achieves near-perfect accuracy (0.98), highlighting its efficacy in transferring complex reasoning capabilities across linguistic barriers.

2 Related Work

Our research is positioned at the intersection of three key areas: temporal reasoning, cross-lingual transfer for low-resource languages, and architectures that decouple reasoning from linguistic form.

Temporal Reasoning in NLP. Temporal reasoning is a long-standing challenge in natural language processing. Early works often relied on rule-based systems and feature engineering to interpret time expressions (Allen, 1983; Setzer and Gaizauskas, 2000; Mani and Wilson, 2000; Pustejovsky et al., 2003). More recently, the advent of pretrained language models has led to significant progress, with benchmarks like TimeBench (Chu et al., 2024) driving the development of sophisticated neural architectures. Models are now commonly fine-tuned on large, task-specific datasets to handle complex temporal queries (Tan et al., 2023). However, this paradigm is heavily dependent on the availability of extensive annotated data, a bottleneck that severely limits performance in low-resource languages like Vietnamese.

Cross-Lingual Transfer for Low-Resource NLP. To overcome data scarcity, various cross-lingual transfer techniques have been proposed. A standard approach is to fine-tune multilingual pretrained models such as mT5 (Xue et al., 2021) or XLM-R (Conneau et al., 2020) directly on the target low-resource language data. Another family of methods involves knowledge distillation, where a large teacher model (typically trained on a high-resource language) transfers its knowledge to a smaller student model for the target language. This has been explored in complex reasoning tasks, such as cross-lingual temporal knowledge graph reasoning, where models learn to align temporal facts across languages (Wang et al., 2023a). Other work focuses on enhancing a model's inherent multilingual capabilities at inference time without retraining, for instance, through test-time scaling techniques (Zhang et al., 2025). While effective, these methods often attempt to embed both linguistic understanding and reasoning capabilities within a single monolithic model, which can be inefficient when the core reasoning logic is language-agnostic.

Decoupling Reasoning from Language. The principle of separating a model's reasoning process from its language-specific surface form is not new and has proven effective in various domains. A prominent example is the evolution from Chain-of-Thought (CoT) prompting (Wei et al., 2023), which generates natural language reasoning steps, to the Program-of-Thought (PoT) paradigm (Chen et al., 2023), which prompts large language models to generate code as an intermediate reasoning step. By offloading the logical computation to a deterministic code interpreter, PoT separates the complex reasoning from the final natural language generation, leading to more robust and interpretable results.

Our work is philosophically aligned with this decoupling principle. However, instead of generating a formal programming language, we use a canonical natural language (English) as the intermediate representation for reasoning. Our hybrid architecture treats the powerful, English-trained Flan-T5 model as a specialized reasoning engine that is agnostic to the original query language. The lightweight Qwen2.5-1.5B-Instruct adapter acts as a flexible semantic parser, translating the intent of the Vietnamese query into a format the engine can execute. This modular approach allows us to effectively plug in a high-resource reasoning capability to a low-resource language without needing to retrain the core engine, demonstrating a practical and efficient method for cross-lingual knowledge transfer.

3 Method: A Decoupled Cross-Lingual Architecture

Our method is structured around a two-stage pipeline where distinct models handle specific sub-tasks: semantic parsing/generation and core temporal reasoning. The overall architecture is depicted in Figure 1.

3.1 The Temporal Reasoning Core (Flan-T5)

At the heart of our system lies a Flan-T5-base model (Chung et al., 2022), which serves as the dedicated reasoning engine. Following the methodology validated by (Tan et al., 2023), we perform Supervised Fine-Tuning (SFT) on a large-scale English dataset of 400,000 samples. This dataset is synthesized using three canonical question patterns:

1. What is the time x year(s) and y month(s) before/after t?
2. What is the time x year(s) before/after t?
3. What is the time y month(s) before/after t?

The model was fine-tuned for **7 epochs** with optimized hyperparameters, endowing it with

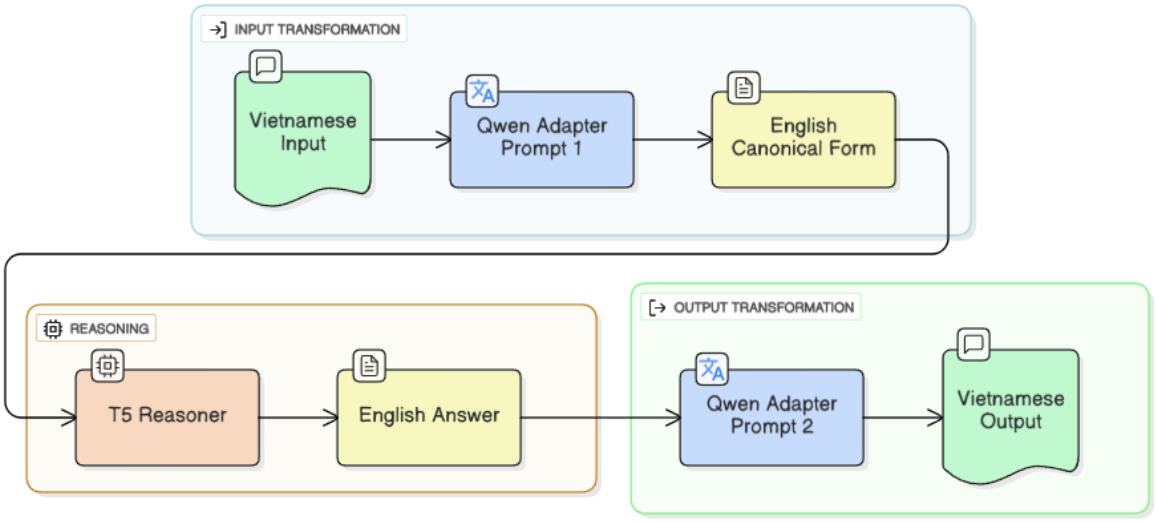

Figure 1: The proposed hybrid architecture. A Vietnamese question is first transformed into a canonical English format by the Qwen2.5-1.5B-Instruct Adapter. The Flan-T5 Reasoner processes this query to produce an English answer, which is then translated back to Vietnamese by the same adapter.

a robust and generalized capability for high-precision date arithmetic, independent of any specific natural language's surface form.

3.2 The Cross-Lingual Semantic Adapter (Qwen2.5-1.5B-Instruct)

To bridge the gap between Vietnamese user queries and our English-centric reasoning core, we deploy a Qwen2.5-1.5B-Instruct model (Bai et al., 2023) as a lightweight, highly-capable adapter. This model performs two critical functions, guided by carefully engineered prompts. To validate the reliability of this crucial component, we evaluated the adapter on the full 500-sample test set. As shown in Table 1, the adapter achieved 100% accuracy on both transformation tasks, ensuring that any final system errors do not originate from the linguistic interface.

Validation of the Adapter. To rigorously and independently evaluate the adapter's accuracy, we created a gold standard evaluation set derived from the 500-sample public test set.

For the Vietnamese-to-English task (Prompt 1), we manually translated each of the 500 Vietnamese questions into its ideal canonical English counterpart. The adapter's output for each question was then compared against this manually created gold standard.

For the English-to-Vietnamese task (Prompt 2), the ground truth was the original Vietnamese answer from the test set. To create the corresponding test inputs, we manually generated the canonical English equivalent of each ground truth answer (e.g., converting *"Tháng 4, 1296"* into *"April, 1296"*). This process allowed us to test the adapter's translation capability in a controlled manner, independent of the T5 reasoner's actual outputs during inference. The adapter's generated text was then compared against the original Vietnamese answers to calculate accuracy.

Table 1: Evaluation results of Prompt 1 (Vietnamese → English) and Prompt 2 (English → Vietnamese).

Prompt	Task	Accuracy
Prompt 1	Vi → En	100%
Prompt 2	En → Vi	100%

Prompt 1: Input Transformation (Vietnamese → English). This prompt instructs the adapter to parse a natural Vietnamese question, extract the key temporal entities (years, months, direction, and base date), and reformulate them into one of the three canonical English patterns understood by the Flan-T5 core.

> **Role:** You are a **canonical time normalizer** that specializes in Vietnamese-to-English transformation of month-year expressions.

Instruction: Parse a Vietnamese question containing a month and year, then reformulate it into one of the three canonical English patterns. Only handle month + year expressions.

Constraints:

- **DO** output only in English.
- **DO** use one of the following canonical patterns:
  1. What is the time X before [Month], [Year]?
  2. What is the time X after [Month], [Year]?
  3. What is the time of [Month], [Year]?
- **DO** write month names in English (January–December).
- **DO** keep year in digits (e.g., 1297, 2020).
- **DO NOT** invent or omit information.

Few-shot Examples:

Input: "Tháng 6, 1297 là khi nào?"
Output: "What is the time of June, 1297?"

Input: "Tháng 4, 2020 là khi nào?"
Output: "What is the time of April, 2020?"

Wrong: "When was June, 1297?"
Correct: "What is the time of June, 1297?"

Reasoning: First, internally map the Vietnamese month-year expression into English. Then, select the closest canonical pattern. Finally, output only the canonical English question. *Do not reveal reasoning, only show the final output.*

Prompt 2: Output Transformation (English → Vietnamese). Conversely, this second prompt handles the final output generation. It instructs the adapter to take the Flan-T5 core's canonical English answer and translate it back into the strict, numeric Vietnamese format required for the final output.

Role: You are a **strict time translator** that outputs month-year expressions in Vietnamese with numeric month format only.

Instruction: Convert the English month-year answer into Vietnamese, following the strict numeric month format.

Constraints:

- **DO** output only in Vietnamese.
- **DO** format strictly as "Tháng [number], [year]".
- **DO** write month as a number (1–12).
- **DO** preserve year exactly.
- **DO NOT** spell out month (e.g., "Tháng Tư").
- **DO NOT** add extra words ("năm", "là", etc.).

Few-shot Examples:

Input: "April, 1296"
Output: "Tháng 4, 1296"

Input: "December, 2020"
Output: "Tháng 12, 2020"

Wrong: "Tháng Tư năm 1296"
Correct: "Tháng 4, 1296"

Reasoning: First, internally map the English month name into its numeric form. Then, format as "Tháng [number], [year]". *Do not reveal reasoning, only show the final output.*

3.3 Inference Pipeline

During inference, the system operates as follows:

1. A Vietnamese question is passed to the Qwen2.5-1.5B-Instruct adapter with **Prompt 1**.
2. The resulting canonical English question is fed into the fine-tuned Flan-T5 reasoning core.
3. The Flan-T5 model computes and returns the answer in English.
4. This English answer is sent back to the Qwen2.5-1.5B-Instruct adapter with **Prompt 2** for the final translation into Vietnamese.

4 Experiments and Analysis

4.1 Dataset

Our experiments utilize datasets from the VLSP 2025 challenge on Temporal Question Answering. We use a large-scale English corpus to train the reasoning core and a Vietnamese corpus for evaluating the final system and training the baselines.

English Reasoning Dataset. To train our Flan-T5 reasoning core, we employ the large-scale English dataset TempReason-L1, introduced in (Tan et al., 2023). This dataset contains **400,000 synthetically generated samples** designed specifically for the Date Arithmetic task. The samples are constructed based on three canonical patterns, ensuring comprehensive coverage of date calculations involving years and months. The scale and controlled structure of this dataset are ideal for teaching the model the underlying logic of temporal arithmetic, independent of complex linguistic variations.

Vietnamese Evaluation Dataset. For our primary evaluation, we use the official Vietnamese dataset from the same VLSP 2025 Temporal QA shared task. This dataset is split into a **training set of 3,000 samples** and a **public test set of 500 samples**. According to the task organizers, this dataset was created by translating and then applying rule-based augmentation to an English seed dataset, resulting in linguistically diverse question patterns in Vietnamese.

In our experiments, the 500-sample test set is used to evaluate all methods. The 3,000-sample training set is used exclusively for fine-tuning the baseline models (e.g., direct Flan-T5 SFT) to establish a fair and challenging point of comparison. Notably, our proposed hybrid architecture does not require this Vietnamese training data, highlighting its effectiveness in a low-resource setting.

4.2 Evaluation Metric and Baselines

We use **Accuracy** as the primary metric, defined as the percentage of predictions that exactly match the ground-truth answer. To provide a comprehensive and rigorous evaluation, we compare our hybrid architecture against a wide range of strong baselines, categorized as follows:

Direct Supervised Fine-Tuning (SFT). This category represents the standard approach of fine-tuning a pretrained model on the 3,000-sample Vietnamese training set.

- **Flan-T5-base** (Chung et al., 2022) (SFT): An instruction-tuned model based on the T5 architecture. This baseline represents a powerful, general-purpose model and tests whether its broad task-solving abilities can be adapted to our specific reasoning task with limited data.
- **mT5-base** (Xue et al., 2021) (SFT): A multilingual T5 model designed for cross-lingual tasks. This baseline tests the effectiveness of multilingual pretraining.
- **ViT5-base** (Phan et al., 2022) (SFT): A T5 architecture extensively pretrained on a large corpus of Vietnamese text. This baseline is crucial as it represents a strong, language-specific SFT approach, testing whether a model with deep prior knowledge of Vietnamese can master the task with the limited training data.

Large Language Models (LLMs). We evaluate a suite of powerful LLMs to establish the state-of-the-art for this task. The models are tested in various configurations:

- **Few-shot Setting:** The model is given five examples of Vietnamese questions and answers before being presented with the test queries. This applies to GPT-OSS-20B and the Gemini series.
- **Advanced Prompting:** For GPT-OSS-20B, we also test an enhanced setting using Chain-of-Thought (CoT) (Wei et al., 2023) prompting combined with Self-Consistency (Wang et al., 2023b) to maximize its reasoning capabilities.
- **Model Variants:** We test multiple versions of the Gemini family, including the lightweight Gemini-1.5/2.5 Flash and the most powerful Gemini-2.5 Pro, to understand the impact of model scale and architecture.

4.3 Results and Analysis

The experimental results, presented in Table 2, provide a comprehensive view of the

Table 2: Performance comparison on the VLSP 2025 public test set. Our hybrid approach matches state-of-the-art LLM performance while significantly outperforming direct fine-tuning methods.

Method	**Accuracy**
Direct Fine-Tuning on Vietnamese Data	
Flan-T5-base (SFT)	0.056
mT5-base (SFT)	0.132
vit5-base (SFT)	0.886
Large Language Models	
GPT-OSS-20B	0.95
GPT-OSS-20B (CoT + Self-consistency)	0.97
Gemini-1.5 Flash	0.95
Gemini-2.5 Flash	0.97
Gemini-2.5 Pro	**0.98**
Ours: Flan-T5 Reasoner + Qwen Adapter	**0.98**

performance landscape for this task and highlight the effectiveness of our decoupled architecture.

Our analysis uncovers a three-tiered hierarchy of performance. At the lowest tier, models that lack deep, specialized pretraining for either the language or the task fail to generalize from the small dataset. The powerful instruction-tuned Flan-T5-base, despite its strong general problem-solving capabilities, collapses completely (0.056). This indicates that its broad, task-agnostic abilities do not effectively transfer to a niche logical reasoning task with such sparse training data. Similarly, the generic multilingual mT5-base fares only slightly better (0.132), as its wide but shallow knowledge across many languages is also insufficient to master the procedural rules of temporal arithmetic.

In the second tier, we observe the power of language-specific pretraining. ViT5-base, which has been extensively pretrained on Vietnamese, achieves an impressive accuracy of **0.886**. This strong result demonstrates that deep familiarity with the target language's syntax and semantics allows a model to learn the task far more effectively from the small training set. However, despite its strength, even this specialized model does not reach the highest level of performance, suggesting that mastering the abstract logic of temporal arithmetic remains a distinct challenge beyond linguistic fluency.

The top tier of performance, at **0.98 accuracy**, is achieved by two distinct approaches. On one hand, the state-of-the-art is set by Gemini-2.5 Pro, a massive, general-purpose proprietary model leveraging its vast emergent reasoning capabilities. On the other hand, our hybrid method successfully **matches this peak performance**. This finding is the central contribution of our work. It demonstrates that our specialized architecture—combining a focused reasoning engine with a lightweight linguistic adapter—is a highly efficient and effective alternative for achieving state-of-the-art results. Instead of relying on the sheer scale of an LLM (like Gemini) or extensive in-language pretraining (like ViT5), our method strategically transfers robust reasoning knowledge from a high-resource language, proving that intelligent system design can be just as powerful as brute-force scale.

4.4 Error Analysis

A manual analysis of the 10 failing cases reveals the specific limitations of our system. As the Qwen2.5-1.5B-Instruct adapter achieved 100% accuracy, all identified errors originate from the Flan-T5 reasoning core. Our investigation classifies these failures into two primary categories based on their magnitude and nature.

Minor Calculation Errors. The most common type of genuine error was a minor miscalculation resulting in an answer that was off by exactly one month. For example, for the query *"Ngày tháng*

nào sẽ là 7 năm sau tháng 11, 1886?" (What is the time 7 years after November, 1886?), the correct answer is *November, 1893*. Our model produced *December, 1893*. This suggests a subtle boundary-condition error in the model's learned arithmetic.

Major Reasoning Failures. We observed rare instances where the model's reasoning process broke down, leading to a wildly inaccurate result. For the query *"Ngày tháng nào sẽ là 6 năm 10 tháng trước tháng 7, 1318?"* (What is the time 6 years and 10 months before July, 1318?), the model produced *September, 1259* instead of the correct *September, 1311*. Interestingly, the temporal offset itself ("6 years and 10 months") is not exceptionally large. A closer look reveals a form of **compositional failure**: the model correctly executes the complex month arithmetic that involves borrowing from the year (July - 10 months $\rightarrow$ September of the previous year), but then fails catastrophically during the subsequent year calculation. This type of error, while infrequent, points to a deeper breakdown in the model's ability to reliably chain multiple procedural steps together.

In conclusion, our analysis indicates that the model's failures are not random but fall into two distinct patterns: small-scale errors at calculation boundaries and large-scale breakdowns in multi-step reasoning. This highlights that future work should focus not just on general accuracy, but specifically on enhancing the numerical robustness and procedural consistency of the reasoning core.

5 Conclusion and Future Work

In this paper, we introduced a novel hybrid architecture that effectively addresses the challenge of temporal reasoning for the low-resource Vietnamese language. By decoupling the core reasoning mechanism from the language-specific interface, our method successfully transfers knowledge from a high-resource language (English) to a low-resource one. Our experiments demonstrate that our decoupled approach is remarkably effective. It not only overcomes the limitations of generic fine-tuning methods but also significantly surpasses strong, language-specific models like ViT5. Crucially, our architecture achieves an accuracy of 0.98, matching the state-of-the-art performance set by the most powerful general-purpose LLMs.

Our key contribution is the empirical validation that a specialized, modular system can be as robust and accurate for specific reasoning tasks as end-to-end, general-purpose models, representing a highly efficient and targeted approach to achieving peak performance. The error analysis further reveals that the system's few remaining weaknesses are logical, not linguistic, opening a clear path for targeted improvement.

For future work, we plan to enhance our system in three main directions. First, we will augment the English training data with challenging edge cases, such as leap year calculations and large-magnitude offsets, to improve the reasoning core's robustness. Second, while our prompt-based adapter is highly effective, we will explore fine-tuning it using parameter-efficient techniques like LoRA (Hu et al., 2021) to create a more compact and specialized translation component. Finally, we aim to extend this decoupled framework to more complex, multi-step temporal reasoning tasks and conduct a human evaluation to assess the naturalness and real-world applicability of its outputs.

References

James F. Allen. 1983. Maintaining knowledge about temporal intervals. *Commun. ACM*, 26:832–843.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, and 29 others. 2023. Qwen technical report. *Preprint*, arXiv:2309.16609.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. 2023. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *Preprint*, arXiv:2211.12588.

Zheng Chu, Jingchang Chen, Qianglong Chen, Weijiang Yu, Haotian Wang, Ming Liu, and Bing Qin. 2024. TimeBench: A comprehensive evaluation of temporal reasoning abilities in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1204–1228, Bangkok, Thailand. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha

Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, and 16 others. 2022. Scaling instruction-finetuned language models. *Preprint*, arXiv:2210.11416.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Inderjeet Mani and George Wilson. 2000. Robust temporal processing of news. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 69–76, Hong Kong. Association for Computational Linguistics.

Long Phan, Hieu Tran, Hieu Nguyen, and Trieu H. Trinh. 2022. ViT5: Pretrained text-to-text transformer for Vietnamese language generation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Student Research Workshop*, pages 136–142, Hybrid: Seattle, Washington + Online. Association for Computational Linguistics.

James Pustejovsky, José Castaño, Robert Ingria, Roser Saurí, Rob Gaizauskas, Andrea Setzer, Graham Katz, and Dragomir Radev. 2003. Timeml: Robust specification of event and temporal expressions in text. pages 28–34.

Andrea Setzer and Robert Gaizauskas. 2000. Annotating events and temporal information in newswire texts. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC'00)*, Athens, Greece. European Language Resources Association (ELRA).

Qingyu Tan, Hwee Tou Ng, and Lidong Bing. 2023. Towards benchmarking and improving the temporal reasoning capability of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14820–14835, Toronto, Canada. Association for Computational Linguistics.

Ruijie Wang, Zheng Li, Jingfeng Yang, Tianyu Cao, Chao Zhang, Bing Yin, and Tarek Abdelzaher. 2023a. Mutually-paced knowledge distillation for cross-lingual temporal knowledge graph reasoning. *Preprint*, arXiv:2303.14898.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023b. Self-consistency improves chain of thought reasoning in language models. *Preprint*, arXiv:2203.11171.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Qiyuan Zhang, Fuyuan Lyu, Zexu Sun, Lei Wang, Weixu Zhang, Wenyue Hua, Haolun Wu, Zhihan Guo, Yufei Wang, Niklas Muennighoff, Irwin King, Xue Liu, and Chen Ma. 2025. A survey on test-time scaling in large language models: What, how, where, and how well? *Preprint*, arXiv:2503.24235.