Discourse Graph Guided Document Translation with Large Language Models

Viet-Thanh Pham, Minghan Wang, Hao-Han Liao, Thuy-Trang Vu
Department of Data Science & AI, Monash University
{thanh.pham1,minghan.wang,hlia0034@student,trang.vu1}@monash.edu

Abstract

Adapting large language models to full document translation remains challenging due to the difficulty of capturing long-range dependencies and preserving discourse coherence throughout extended texts. While recent agentic machine translation systems mitigate context window constraints through multi-agent orchestration and persistent memory, they require substantial computational resources and are sensitive to memory retrieval strategies. We introduce TRANSGRAPH, a discourse-guided framework that explicitly models inter-chunk relationships through structured discourse graphs and selectively conditions each translation segment on relevant graph neighbourhoods rather than relying on sequential or exhaustive context. Across three document-level MT benchmarks spanning six languages and diverse domains, TRANSGRAPH consistently surpasses strong baselines in translation quality and terminology consistency while incurring significantly lower token overhead.

1 Introduction

Recent advances in large language models (LLMs) have revolutionised machine translation (MT). LLM-based MT models have achieved remarkable performance at the sentence level and demonstrated promising capabilities for document-level translation (DocMT), often surpassing specialised encoder-decoder MT systems (Xu et al., 2024; Lyu et al., 2024; Wu et al., 2024a; Pang et al., 2025). Despite these successes, translating entire long documents remains a challenge, particularly in the modelling of long-range dependencies and discourse phenomena across sentences and paragraphs.

Fine-tuning LLMs on high-quality document-level data has improved multilingual contextualisation, yet the limited context window poses a persistent bottleneck (Li et al., 2025; Ramos et al., 2025). Recent agentic MT approaches address this by coordinating multiple specialised LLM agents with persistent memory to store and retrieve discourse information, such as proper nouns, bilingual summaries and short/long-term cache, throughout translation (Wu et al., 2024b, 2025; Wang et al., 2025; Dutta et al., 2025). These systems achieve impressive quality and style control, including for literary text, but often at high orchestration and token costs, with quality sensitive to memory hygiene and retrieval placement. Meanwhile, other research suggests that structured and selective context—rather than *more* context—can be key: well-chosen cross-sentential signals and explicit discourse cues (coreference, cohesion devices) can produce document-level gains without complex pipelines (Zhang et al., 2022; Qin et al., 2022).

In this paper, we take a discourse-guided, *selective-context* approach and propose TRANSGRAPH, a two-stage, graph-conditioned procedure for document-level MT. Rather than relying on exhaustive sequential context or memory-intensive retrieval systems, we first partition a document into coherent chunks and construct a labelled inter-chunk graph that explicitly encodes discourse relations frequently driving translation choices, such as *Entity-Coreference*, *Core→Detail*, *Motivation→Method*, drawing on Rhetorical Structure Theory and modern discourse parsing insights (Mann and Thompson, 1988; Taboada and Mann, 2006; Jiang et al., 2023b). During translation, each chunk is conditioned not on all preceding text, but on a small, discourse-guided neighbourhood, enriched with discourse relation labels. TRANSGRAPH offers three key advantages: (i) selective conditioning only on the contextual information that matters for the translation; (ii) improving terminology and cohesion with minimal token overhead; and (iii) remaining fully agnostic to the underlying model or backbone architecture.

We evaluate TRANSGRAPH on three DocMT benchmarks featuring tagged terminologies across diverse domains, including technical presentations

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics
Volume 1: Long Papers, pages 1607–1627
March 24-29, 2026 ©2026 Association for Computational Linguistics

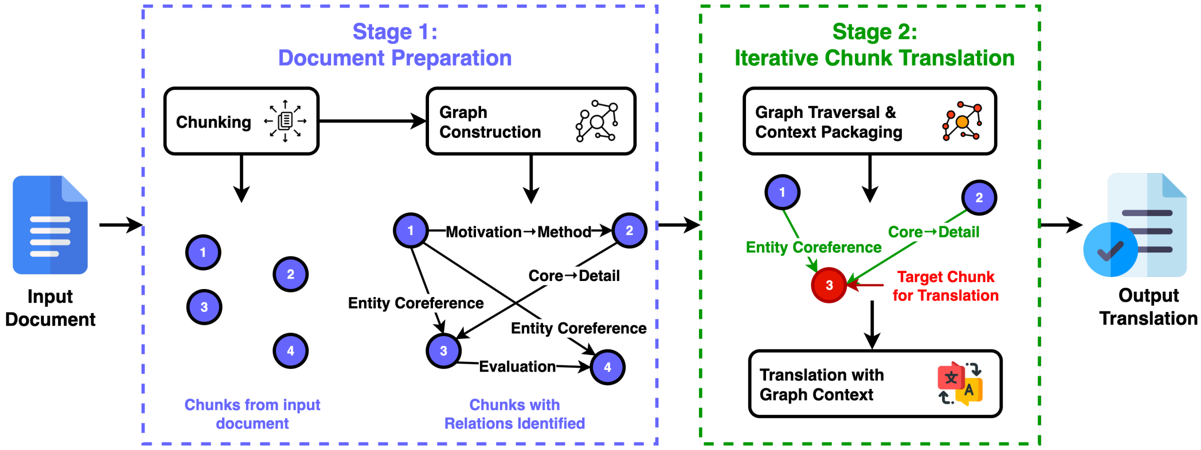

Figure 1: Illustration of our proposed framework, *TransGraph*. In Stage 1, *TransGraph* takes a document as input, split it into multiple small chunks and identify the discourse relations between every pair of chunks and represent them as a knowledge graph. In Stage 2, the translation of one chunk proceeds by retrieving adjacent chunks in the graph with their corresponding relations.

and literary texts, spanning six languages. Our method consistently outperforms sentence-level baselines, sequential-context DocMT, and agentic MT systems in both document-level BLEU and COMET scores, as well as terminology accuracy while incurring substantially lower token costs. In summary, our contributions are:

- A discourse graph guided DocMT framework that injects *relation-labelled* document context, aligning with discourse theory while remaining lightweight and backbone-agnostic.
- We demonstrate that structured discourse relations outperform generic sequential context expansion, achieving consistent improvements in translation quality.
- Extensive comparisons to agentic MT systems, highlighting a strong quality–cost tradeoff and robustness under long documents.

2 Document Translation with Discourse Graph

We present TRANSGRAPH, a two-stage, graph-conditioned procedure for document-level machine translation, illustrated in Figure 1. In Stage 1, TRANSGRAPH first splits the source document into coherent textual units (chunks) and construct a directed graph of discourse-level relations between chunks. In Stage 2, the framework then performs translation while conditioning each chunk on graph-retrieved context. We denote $V = \{c_1, \ldots, c_N\}$ the set of chunks and $G = (V, E, \ell)$ a directed, labeled graph representing the discourse relations among pairs of chunks, where $E \subseteq V \times V$ and $\ell : E \to \mathcal{R}$ assigns a label for an edge from a predefined schema, capturing discourse-level relations such as *Entity-Coreference*, *Core→Detail*, *Motivation→Method*, and *Evaluation*. Below sections will discuss each step in the procedure.

2.1 Stage 1: Document Preparation

The objective of Stage 1 is to produce coherent chunks and especially an interpretable discourse graph that captures salient inter-chunk relations within a bounded window, thereby providing structured context for downstream translation.

Chunking The document is segmented into contiguous chunks that preserve sentence boundaries. We iteratively loop through T tokens of the source document and prompt the LLM to split the text into chunks (Prompt template shown in Appendix A, Listing 1). These chunks are then appended to the chunk set V and the LLM continues to process the next T tokens. In case the last n tokens of T do not end with a punctuation, we merge them with the next T tokens and ask LLM to perform chunking again. This process ensures that (i) we avoid the long-context problem of LLM by providing it with limited T tokens and (ii) we preserve strict sentence boundaries.

Graph Construction and Relation Labeling Inter-chunk structure is represented as a directed, labeled graph $G = (V, E, \ell)$. To control computational cost while preserving locality, we only identify relations of forward pairs within a small window $\mathcal{P}$: $\mathcal{P} = \{(i, j) : 1 \le i < j \le N,\ j - i \le w\}$.

where w is the window size, (i, j) are the indexes of chunk i and j. For each $(i, j) \in \mathcal{P}$, the LLM is prompted with the chunk pair (c_i, c_j) to decide whether a meaningful relation holds, its direction, and a label $\ell(c_i, c_j) \in \mathcal{R}$ for each edge (Prompt template in Appendix A, Listing 2).

Based on prior works and theories on rhetorical structure (Mann and Thompson, 1988; Taboada and Mann, 2006; Jiang et al., 2023b), we define 10 types of relation for the LLM to choose from. Besides several existing types of relation (e.g. *Motivation→Method*, *Cause→Effect*, and *Contrast*), we also added several relation types focusing on ensuring coherent translation and terminology consistency, such as *Entity-Coreference* and *Terminology Definition*. All relation types in our setup are described in Section 2.1. Pairs judged as *No-Relation* are discarded.

2.2 Stage 2: Iterative Chunk Translation

Given (V, G), translation proceeds by traversing the graph, assembling compact context for each target chunk from its in-neighborhood, and prompting the LLM to produce the target-language translation with the assembled context. The stage is iterative in the sense that the procedure is repeated for each chunk. By conditioning each translation step on the appropriate discourse relations to neighboring contents, the system improves referent resolution, terminology consistency, and rhetorical coherence.

Graph Traversal Chunks are processed in a stable index order, consistent with the document flow. For a target chunk c_j for translation, we gather its immediate in-neighbors in the graph $\mathcal{N}^-(j) = \{\, i \mid (c_i, c_j) \in E \,\}$. This choice aligns with the causal direction of discourse found in natural documents: antecedents typically precede anaphors, motivations precede methods, and core statements precede elaborations (Mann and Thompson, 1988), therefore leading to better consistency in translation.

Context Packaging For each chunk index j, we assemble a structured context package $\mathcal{C}_j$ comprised of a small set of formatted records from $\mathcal{N}^-(j)$: $\mathcal{C}_j = \{(i,\, c_i,\, \ell(c_i, c_j)) : \ i \in \mathcal{N}^-(j)\}$. Each record includes the neighbor chunk content and its relation label, enabling the translator to treat evidence according to its discourse role rather than as undifferentiated context (Example provided in Appendix). To avoid the long-context problem of LLMs, $|\mathcal{N}^-(j)|$ is capped; ties are resolved by preferring earlier neighbors (e.g. pair of chunks that are adjacent to each other), which empirically benefits term consistency and document coherence.

Translation with Graph Context The translation of c_j is produced by prompting an LLM with the pair $(c_j, \mathcal{C}_j)$. The prompt (given in Appendix A, Listing 3) instructs the model to preserve terminology and discourse relation specified by each relation label $\ell(c_i, c_j)$. Denoting the translation operator realized by the LLM as $T(\cdot)$, the output is represented as $\hat{y}_j = T(c_j \,|\, \mathcal{C}_j)$. After all chunks are translated, the target document is reconstructed by concatenation in source order: $\hat{Y} = \hat{y}_1 \,\|\, \cdots \,\|\, \hat{y}_N$. Because each translation is conditioned on its in-neighborhood, boundary artifacts are mitigated without imposing expensive global decoding constraints.

3 Experiments

3.1 Experimental Setups

We conduct experiments with different open-source families of LLMs, including Qwen3 (Yang et al., 2025), Llama3 (Grattafiori et al., 2024) and Ministral[1]. Decoding parameters, such as temperature, top-k, for each model are set to the recommended settings from the authors of these LLMs. Regarding the hyperparameter choices, we empirically set T in Stage 1 (Chunking) to be 100 tokens and window size w to be 20, followed the common practice of graph-based retrieval works (He et al., 2024; Kim et al., 2023), whereas the number of in-neighbour chunks $\mathcal{N}^-(j)$ to retrieve in Stage 2 is set to 5 chunks.

To demonstrate the performance and efficiency of TRANSGRAPH, we compare our proposed framework with the following baselines:

- **Sentence-Level Translation (SENT. MT)** We prompt LLM to iteratively translate each sentence in the source documents.
- **Single-Pass Document Translation (1-PASS DOCMT)** We prompt LLM to translate the whole source document in a single run.
- **TransAgent (Wu et al., 2025)** - a multi-agent framework designed for document translation. It incorporates several steps to help LLMs

[1] Model signatures: `Qwen/Qwen3-8B`, `Qwen/Qwen3-14B`, `Qwen/Qwen3-32B`, `meta-llama/Llama-3.1-8B-Instruct` and `mistralai/Ministral-8B-Instruct-2410`.

Relation (value)	Typical direction	Definition
Background→Core	forward	i supplies necessary context that enables understanding claims or results in j.
Core→Detail	forward	j elaborates on or narrows a point introduced in i (part–whole, attribute, instance).
Problem→Solution	forward	i states a need, gap, or problem; j presents a method or solution addressing it.
Cause→Effect	forward	i presents a cause or reason; j describes the resulting outcome.
Contrast	either (often symmetric)	i and j assert opposing or divergent claims, properties, or outcomes.
Comparison	either (often symmetric)	i and j compare entities along shared dimensions without opposition.
Condition	forward (common)	i states a condition or hypothesis; j describes what holds under that condition.
Evaluation	forward	j evaluates, critiques, or validates the approach introduced in i.
Entity Coreference	either (symmetric)	i and j mention the same entity (dataset, model, variable) establishing continuity.
Terminology Definition	forward (common)	i defines a term that appears or is used in j.
No relation	N/A	No meaningful topical or semantic connection between i and j.

Table 1: Discourse-level relations used for knowledge graph construction. Direction: *forward* means earlier Chunk i → later Chunk j; *either* is either Chunk i → Chunk j or Chunk j → Chunk i are acceptable.

maintain consistency while translating long-form text, such as summarization, glossary construction, and tone and style detection.

- **DeLTA (Wang et al., 2025)** - an agentic framework that focus on maintaining translation consistency. It features a multi-level memory structure that stores information across various granularities and spans, including proper noun records, bilingual summary, long-term memory, and short-term memory, which are continuously retrieved and updated by auxiliary LLM-based components.

- **TransGraph- Fixed Size Chunking (Fixed)** We remove the chunking component of TransGraph and use fixed-size chunking. Specifically, for a given input document, we split the document into 10 length-equivalent chunks, based on the number of tokens.

- **TransGraph- Removing Discourse Relations (−Rel)** Instead of iteratively translate chunks conditioning on the neighboring chunks and their discourse relations, we take 5 previous chunk contents as context for the translation of the current chunk.

- **TransGraph- Conditioning on Sequential Context (Seq)** We modify the Context Packaging step of TransGraph, in which we only select chunk i for translating the current chunk j only if i is in the range $[j-1, j-5]$.

Benchmarks To assess the performance of translation methods, we select three document translation benchmarks. Among these three, two benchmarks have tagged terms in both source and target documents, from which we can calculate the accuracy of translation methods in maintaining terminology accuracy. Table 2 shows the statistics of each dataset.

Dataset	ACL 60/60	BWB	GuoFeng
Total # of Docs	5	80	25
Avg # of Sentences	84.20	39.91	57.86
Max # of Sentences	101	47	143
Min # of Sentences	57	19	28
Avg # of Tokens	1746.40	945.41	1828.36
Max # of Tokens	2031	1801	4046
Min # of Tokens	1519	717	817

Table 2: Dataset statistics.

- **ACL 60/60 (Salesky et al., 2023).**The first is ACL 60/60, which features 5 technical presentations from ACL 2022, each of which comes with an English talk and 10 corresponding translations to 10 languages. ACL 60/60 includes tags indicating NLP-related technical terms in the source and target talks, which we will use to evaluate terminology accuracy of translation methods. We consider 12 translation directions for the experiments: En ⇔ Zh, De, Ja, Fr, Pt, and Ru.

- **BWB (Jiang et al., 2023a).** The BWB dataset is a large-scale document-level Chinese-English parallel dataset. It consists of 80 Chinese documents across multiple genres (sci-fi, romance, action, fantasy, comedy, etc.) and their corresponding English translations. Like ACL 60/60, this dataset also has tagged terminologies available. We consider 2 translation directions with the BWB benchmark: En ⇒ Zh and Zh ⇒ En.

- **Guofeng Webnovel (Wang et al., 2024b).** The Guofeng Webnovel includes high-quality documents and translations in English and Chinese. We use the TEST_1 and TEST_2 subsets of the benchmark, which accumulate 25 documents for evaluation. We consider 2 translation directions with the Guofeng Webnovel: En ⇒ Zh and Zh ⇒ En.

Evaluation Metrics We opt for the classic document-level BLEU (d-BLEU) and document-level COMET (d-COMET) (Vernikos et al., 2022). Regarding the terminology-focused benchmarks (BWB and ACL 60/60), which have labeled terms in both source and reference documents, we add Terminology Accuracy as another evaluation metric, which computes the accuracy of translating terminologies when compared with terms in reference documents.

3.2 Experiment Results

Across all backbones and datasets, TRANSGRAPH delivers consistent gains over both sentence-level and single-pass document baselines, and is competitive with, often superior to, recent agentic systems.

Regular Document Translation. On the combined GuoFeng test sets (Table 3), TRANSGRAPH improves d-BLEU and d-COMET for most LLM backbones. With Qwen3-32B, TRANSGRAPH attains the best overall score, exceeding the sentence baseline by +6.38 BLEU and +9.24 COMET, and edging out TRANSAGENT and DELTA. Trends hold for Llama3.1-8B and Ministral-8B backbones, where TRANSGRAPH yields the highest d-COMET (83.26 and 81.37, respectively). One exception appears with Qwen3-8B, where TRANSGRAPH tops d-BLEU (26.24) but trails DELTA on d-COMET (79.29 vs. 81.80), suggesting that strong local faithfulness and style modelling remain challenging for the smallest backbone when only shallow context is available. Overall, the pattern indicates that graph-conditioned retrieval provides stable quality benefits without expensive and costly multi-agent orchestration.

Terminology-focused Document Translation. Terminology-focused translation is where TRANSGRAPH is most distinctive. As shown in Table 3, on the BWB benchmark, with the Qwen3-32B backbone, TRANSGRAPH exhibits a termninology accuracy of 61.72, outperforming both TRANSAGENT and DELTA, as well as the other two baselines. TRANSGRAPH also showed outstanding performance in terms of terminology with other LLM backbones, including both medium and lightweight models. On the ACL60/60 benchmark, TRANSGRAPH also has better accuracy across different backbones (e.g., Qwen3-32B: 78.65 vs. 74–75 with agentic baselines). These improvements arrive without sacrificing d-BLEU/d-COMET; for instance, on ACL60/60 with Qwen3-32B, d-COMET of TRANSGRAPH reaches 89.78 while terminology accuracy remains in first place, indicating that explicit discourse guided translation helps preserve specialized terms without degrading general adequacy/fluency.

3.3 Discussions

In this section, we conduct various intrinsic experiments of TRANSGRAPH to justify our design choices, as well as perform a cost analysis and cohesion analysis of TRANSGRAPH and other document translation methods.

3.3.1 Chunking Analysis

In this section, we measure the document chunking performance of LLMs by performing an evaluation on the ACL 60/60 benchmark. We first perform a manual annotation to label the position of each chunk for the 5 English documents, then calculate the overlapping rate of chunks split by LLMs in different languages. Specifically, we give the models the documents in different languages available in the ACL 60/60 test set (En, Zh, De, Ja, Pt, and Ru) and calculate the overlapping rate of generated chunks w.r.t. our manual annotation by (i) iterating through each pair of chunks and (ii) computing the ratio of overlapping sentences over the maximum number of sentences between two chunks. Table 4 shows the results. As expected, chunking performance decreases gradually with smaller model sizes, reflecting the translation performance shown in Table 3. En and Zh have the best chunking quality across LLMs, as Qwen-based models are trained heavily on English and Chinese.

3.3.2 Graph Analysis

Relation identification accuracy. We first evaluate the performance of LLMs in determining the relations between chunks to construct the discourse graph. From the ACL60/60 benchmark, we manually annotated 20 samples for each relation type, resulting in a test set of 220 samples. Each sample comes with a pair of chunks in English, Chinese, and German, along with their labeled relation type. Table 5 shows the evaluation results. Qwen3-32B

Method	BWB			ACL 60/60			GuoFeng	
	d-BLEU	d-COMET	Terminology Acc	d-BLEU	d-COMET	Terminology Acc	d-BLEU	d-COMET
Qwen3-32B								
Sent. MT	11.01	46.67	41.99	36.86	81.95	73.83	26.45	81.23
1-Pass DocMT	18.77	83.28	46.47	37.14	81.34	72.79	29.88	89.21
TransAgent	**19.06**	**84.73**	53.21	37.13	81.82	74.69	31.57	90.32
DeLTA	15.53	80.34	52.03	**39.54**	87.66	74.44	31.28	89.98
TransGraph (−Rel)	19.27	83.78	51.05	37.27	83.87	74.55	30.72	89.72
TransGraph (Fixed)	14.92	79.96	60.20	37.55	88.94	78.06	32.54	90.38
TransGraph (Seq)	14.87	79.36	52.47	37.09	88.98	74.41	31.87	90.04
TransGraph (Ours)	15.27	80.26	**61.72**	37.59	**89.78**	**78.65**	**32.83**	**90.47**
Qwen3-14B								
Sent. MT	10.38	43.92	39.61	35.07	80.54	71.88	28.21	87.06
1-Pass DocMT	14.53	78.24	45.42	35.63	79.77	71.95	28.32	88.40
TransAgent	14.44	79.12	52.13	35.52	80.21	73.47	28.86	89.25
DeLTA	17.73	81.92	50.72	36.61	**88.92**	73.82	28.98	**88.58**
TransGraph (−Rel)	16.18	79.55	51.92	36.37	81.67	73.65	29.07	88.02
TransGraph (Fixed)	17.69	82.17	59.15	37.84	85.48	77.04	29.73	88.10
TransGraph (Seq)	17.54	81.71	51.27	37.59	85.31	73.49	28.96	88.05
TransGraph (Ours)	**17.94**	**82.61**	**60.68**	**38.09**	86.11	**77.61**	**30.08**	88.19
Qwen3-8B								
Sent. MT	9.26	41.12	35.94	33.24	78.87	70.63	22.04	82.11
1-Pass DocMT	16.22	78.41	43.11	33.98	78.56	69.92	21.62	81.44
TransAgent	13.14	75.16	50.24	33.72	79.23	72.02	23.46	81.23
DeLTA	12.93	76.62	48.41	35.04	**87.71**	72.69	26.22	**81.80**
TransGraph (−Rel)	16.40	76.10	47.63	34.62	80.52	71.78	20.22	81.61
TransGraph (Fixed)	16.77	78.68	56.68	35.90	84.43	75.51	25.92	79.18
TransGraph (Seq)	16.43	78.21	49.18	35.61	84.28	72.20	25.58	79.11
TransGraph (Ours)	**16.83**	**79.11**	**58.19**	**36.11**	85.08	**76.13**	**26.24**	79.29
Llama3.1-Instruct-8B								
Sent. MT	12.79	73.57	36.22	19.30	0.62	57.15	22.10	81.41
1-Pass DocMT	13.30	73.31	43.58	19.47	0.77	62.33	23.54	82.19
TransAgent	13.78	74.70	50.09	19.98	0.76	62.98	**23.85**	82.45
DeLTA	12.79	**75.78**	48.33	22.33	**0.78**	67.68	22.14	81.32
TransGraph (−Rel)	13.68	73.32	48.13	20.38	0.77	67.05	23.74	83.05
TransGraph (Fixed)	14.45	73.34	57.22	22.21	0.78	76.13	22.95	83.18
TransGraph (Seq)	14.18	72.44	49.06	21.91	0.77	65.18	22.87	83.03
TransGraph (Ours)	**14.58**	73.34	**58.74**	**22.51**	**0.78**	**77.83**	23.07	**83.26**
Ministral-Instruct-8B								
Sent. MT	9.51	75.40	34.87	36.08	0.79	67.67	21.99	79.22
1-Pass DocMT	9.62	71.50	42.15	35.52	0.75	62.55	21.81	80.17
TransAgent	9.83	75.35	49.18	41.40	0.87	77.43	22.65	80.34
DeLTA	**15.39**	74.85	47.60	40.13	0.74	67.70	22.15	81.21
TransGraph (−Rel)	10.99	75.76	46.62	39.22	0.79	68.91	22.37	80.92
TransGraph (Fixed)	13.92	76.27	55.57	46.61	0.87	80.39	22.82	81.28
TransGraph (Seq)	14.01	75.46	48.24	47.24	0.87	72.41	22.71	81.30
TransGraph (Ours)	14.41	**76.36**	**57.06**	47.84	**0.88**	**82.57**	**22.99**	**81.37**

Table 3: Results of different document-level MT methods on terminology-focused benchmarks (BWB, ACL 60/60 benchmark) and regular benchmark (GuoFeng TEST_1 and TEST_2).

Models	Qwen3-8B	Qwen3-14B	Qwen3-32B
Annotated ⇔ En	80.23	87.73	90.96
Annotated ⇔ De	60.62	72.11	81.93
Annotated ⇔ Zh	79.37	84.60	91.72
Annotated ⇔ Ru	64.15	71.94	76.87
Annotated ⇔ Pt	74.63	83.12	88.13
Annotated ⇔ Ja	58.28	63.19	66.46
Annotated ⇔ Fr	65.91	74.32	81.02

Table 4: Average overlaping rate across language pairs and model sizes.

Model	Qwen3-32B	Qwen3-14B	Qwen3-8B
Graph Accuracy			
En-Acc	94.54	90.45	87.27
De-Acc	89.09	88.18	77.72
Zh-Acc	90.45	90.90	80.45
Graph Consistency			
En & De	76.81	75.00	70.45
En & Zh	87.15	86.85	79.18

Table 5: Graph Accuracy and Consistency of different LLM backbones.

achieves 94.54% on English, 89.09% on German, and 90.45% on Chinese accuracy. Accuracy degrades gradually with smaller models (Qwen3-14B around 88–91%; Qwen3-8B around 78–87%), mirroring translation-quality trends. These results are expected, as proven in other reasoning tasks like solving mathematical problems (Yang et al., 2025), smaller models have poorer reasoning capabilities. This sensitivity analysis is important because false relation labels can introduce misleading context; empirically, however, TransGraph's reliance on a bounded in-neighborhood and label-aware prompts appears robust to sporadic mislabeling.

Graph–graph consistency across languages. From the ACL60/60 benchmark, we compare discourse graphs induced from English source documents to graphs induced from Chinese and German references. Graphs are constructed by going through Stage 1 of TransGraph, and Graph Consistency is measured by calculating the overlapping rate of two sets of relations deduced from a pair of discourse graphs in different languages. Qwen3-32B attains 76.81% (En & De) and 87.15% (En & Zh) consistency, while Qwen3-14B is close (75.00/86.85%) and Qwen3-8B remains reasonable (70.45/79.18%). The higher En & Zh consistency likely reflects the raw performance of Qwen3, as this model family is trained primarily on English and Chinese. These findings support our design choices of TransGraph - when the LLM is conditioned on explicit discourse roles (Background→Core, Core→Detail, etc.), rhetorical structure is more faithfully preserved cross-lingually.

3.3.3 Design Choices Justification

We justify the design choice of TransGraph by comparing it with the following baselines: (i) TransGraph (Fixed), (ii) TransGraph (−Rel), and (iii) TransGraph (Seq).

Importance of Coherent Chunking. Table 3 shows that naively performing document chunking with uniform-length chunks (TransGraph (Fixed)) exhibits worse performance compared to TransGraph on all benchmarks. Although the evaluation results of this baseline still consistently outperform other baselines, especially on Terminology Accuracy, it is not optimal since the chunks are not split based on content, leading to unnecessary sentences in chunks, which confuses the LLMs in identifying the discourse relations.

Importance of Discourse Relations. We also experimented with removing the discourse graph and just providing LLMs with the 5 previous chunks' content when translating a target chunk (TransGraph (−Rel)). As shown in Table 3, performance is much worse than TransGraph, and it is only slightly better than 1-Pass DocMT baseline. This proves that discourse relations are important in maintaining translation quality and terminology accuracy.

To further analyze the contribution of discourse relations during the translation process, we perform an additional experiment by first categorizing the relations into the following groups (categories defined in (Mann et al., 1989)):

- **Elaborative Relations:** Background→Core, Core→Detail.
- **Causal Relations:** Cause→Effect, Condition, Problem→Solution.

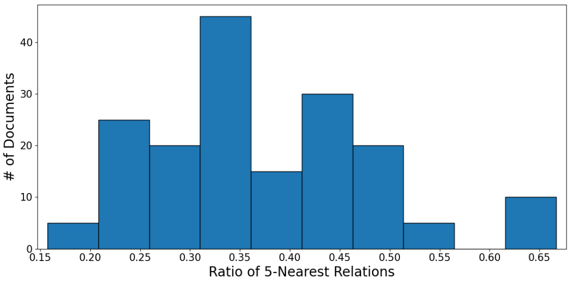

Figure 2: Distribution of ratios of 5-nearest relations out of the total number of relations. Distribution is calculated on the documents of BWB and ACL 60/60 benchmarks.

- **Comparative Relations:** Contrast, Comparison
- **Evaluation Relation:** Evaluation.
- **Coherence & Consistency Relations**: Entity Coreference, Terminology Definition.

With Qwen3-32B as the backbone model, we analyze TransGraph performance when removing each relation group from the set of relations and provide the results in Table 6. As expected, dropping the proposed Coherence & Consistency Relations causes the largest performance drops in terminology-focused benchmarks (BWB and ACL 60/60). The second largest performance drop comes from dropping the Elaborative Relations. This is also expected, as these relations carry important definitions, appositions, and clarifying details in the documents. These results further justified the importance of using discourse relations for aiding reasoning capabilities of LLMs in document translation.

Importance of the Graph. Finally, in order to measure the effectiveness of the graph design in document translation, we compare TransGraph with TransGraph (Seq) baseline . As shown in Table 3, while the d-BLEU and d-COMET scores of this baseline are slightly lower than the original TransGraph, the Terminology Accuracy is significantly worse. This is because when we only condition the translation of a chunk on the nearest chunks, the LLM does not have access to faraway chunks that may directly contribute to the translation of specific terms.

Figure 2 illustrates the ratio distribution of the number of existing relations among 5-nearest chunks compared to the total number of relations in the BWB and ACL 60/60 benchmarks. Most of the documents have low ratios around 35%, which

Method	BWB			ACL 60/60			GuoFeng	
	d-BLEU	d-COMET	Terminology Acc	d-BLEU	d-COMET	Terminology Acc	d-BLEU	d-COMET
TransGraph- w/o Elaborative Relations	14.63	79.37	54.25	35.40	86.23	75.78	31.56	90.02
TransGraph- w/o Causal Relations	14.67	80.04	55.85	37.45	87.12	76.64	31.88	90.13
TransGraph- w/o Comparative Relations	15.02	79.49	56.92	37.48	88.71	76.81	32.09	90.21
TransGraph- w/o Evaluation Relation	15.18	80.14	59.99	37.51	89.30	78.22	32.30	90.28
TransGraph- w/o Coherent Relations	12.66	79.25	51.58	36.32	84.76	74.76	31.04	89.83
TransGraph	**15.27**	**80.26**	**61.72**	**37.59**	**89.78**	**78.65**	**32.83**	**90.47**

Table 6: Results of TransGraph with `Qwen3-32B` backbone when omitting relation groups from the set of relations on terminology-focused benchmarks (BWB, ACL 60/60 benchmark) and regular benchmark (GuoFeng TEST_1 and TEST_2).

means that if we only consider 5-nearest chunks in the graph of TransGraph, we are missing out 65% of the relations, resulting in poorer performance and terminology consistency.

3.3.4 Cost Analysis

We analyze the cost of different document translation methods to highlight the efficiency of TransGraph (Table 8). We report the average number of input & output tokens, as well as the average number of LLM calls and total tokens when running on the three document translation benchmarks. TransAgent and DeLTA incur large budgets due to back-and-forth multi-agent exchanges and sentence-granular passes - average total tokens ≈ 104k and 171k per document, respectively. By contrast, TransGraph averages ≈ 59k tokens with 42 calls on average - substantially below agentic baselines, while maintaining higher accuracy on terminology and stronger BLEU/COMET. As expected, single-pass document translation is cheapest (≈1.8k tokens, 1 call) but underperforms markedly on document-level metrics and terminology. Sentence-level decoding is also cheap per call but accumulates calls (avg 32) and suffers from maintaining cohesion and terminology accuracy. In short, TransGraph occupies a favorable quality–cost regime - one-order-of-magnitude fewer tokens than multi-agent frameworks, yet with stronger document translation performance.

3.3.5 Cohesion Analysis

We further analyzed cohesion in different translation methods on the ACL 60/60 benchmark. Unlike BLEU and COMET, evaluating discourse phenomena like coreference and conjunction requires deep semantic understanding. While existing methods (Tan et al., 2022; Jiang et al., 2023c) assess whether translations preserve entity dependencies and logical relationships, their complex annotation pipelines limit practical use. We simplify this process using LLM-as-a-Judge: an LLM first annotates pronouns with their referents and conjunctions with logical relationships in the source text, then checks translation accuracy in the target text using this annotation as grounding. Unlike other works, which use LLM-as-a-Judge to give a singular score for performance estimation, we decompose cohesion assessment into coreference and conjunction accuracy, making the evaluation much simpler for the LLMs to do, and easier for us to interpret the scores. Specifically, our cohesion evaluation process works as follows. First, we use `Gemini-2.5-Flash` for its strong capabilities and long context length to annotate special words in the source text with XML inline tags (e.g. `[He]<type="personal" referent="John">` for coreferences and `[If]<type="subordinating" relationship="condition">` for conjunctions). Detailed prompt templates are provided in in Appendix A. Next, instead of annotating translations separately (which causes alignment issues due to the diversity of the translation text, thus introducing biases in the scoring), we use the source text as grounding and supplement translation correctness within existing tags: `[his]<type="..." referent="..." translation="seine" correct="true">`. Finally, we calculate accuracy rates from the annotated inline tags as cohesion metrics.

As shown in Table 7, for coreference, doc-level methods significantly outperform sentence-level translation, with TransGraph surpassing baselines across multiple languages and achieving perfect scores in Japanese and Chinese. For conjunction, TransGraph achieves performance comparable to DeLTA, with both agentic and graph-based methods expressing logical relationships more accurately than simple sentence/doc-level approaches. Interestingly, translation systems sometimes score higher than the reference because ACL 60/60 contains spoken language where references retain colloquial errors that reduce readability, while translation systems produce more polished, written-style output, resulting in higher scores.

Dims.	Coreference						Conjunction					
Lang. (En-XX)	Sent-level	Single-Pass Doc-level	TRANSAGENT	DELTA	TRANSGRAPH (ours)	Reference	Sent-level	Single-Pass Doc-level	TRANSAGENT	DELTA	TRANSGRAPH (ours)	Reference
De	80.33	**99.35**	96.03	95.80	97.37	97.68	80.41	75.18	82.10	**82.95**	82.21	87.02
Fr	78.85	98.03	96.81	98.69	**98.06**	97.56	75.87	76.76	76.37	**84.04**	81.32	95.52
Ja	74.88	98.73	97.79	98.08	**100.00**	100.00	63.69	63.27	77.58	76.32	**78.99**	84.28
Pt	74.08	97.22	96.53	97.89	**98.97**	**97.57**	73.51	83.21	85.84	**89.61**	**85.77**	92.91
Ru	79.57	98.11	97.99	97.93	**98.74**	**97.91**	75.35	75.02	80.74	83.50	**83.92**	77.43
Zh	82.65	98.52	**100.00**	99.57	**100.00**	99.56	67.04	72.62	80.54	77.62	**81.51**	80.64
Avg.	78.39	98.33	97.52	97.99	**98.86**	98.38	72.65	74.34	80.53	**82.34**	82.29	86.30

Table 7: Cohesion evaluation on the ACL 60/60 dataset, scores represent translation accuracy on coreference and conjunction with a range between 0 and 100. Best-performed methods within each language pair are **bolded** (dataset reference is excluded).

Method	Avg Input Tokens	Avg Output Tokens	Avg LLM Calls	Avg Total Tokens
SENT. MT	45	36	32	2,592
1-PASS DOCMT	1,010	815	1	1,825
TRANSAGENT	932	251	88	104,104
DELTA	621	24	266	171,304
TRANSGRAPH (Ours)	1,351	98	42	59,368

Table 8: Cost analysis of different DocMT methods.

Doc ID	**111**	**410**
Coreference Annotation Accuracy	97.2	98.8
Coreference Evaluation Accuracy	100.0	100.0
Conjunction Annotation Accuracy	98.0	100.0
Conjunction Evaluation Accuracy	93.9	98.4

Table 9: Accuracy of `Gemini-2.5-Flash` when used as a Judge for cohesion evaluation.

We also carry out a manual evaluation to assess the correctness of `Gemini-2.5-Flash` for this cohesion evaluation task. We manually annotated two documents in the ACL 60/60 benchmark for both the coreference and conjunction dimensions. Results are provided in Table 9. In most cases, we observed that gemini-2.5-flash produces exactly identical annotations as our human annotators - only a few cases are identified to be incorrect. This is expected, as the tasks are easy and require little to no reasoning from the model.

4 Related Works

DocMT and context integration Early DocMT incorporated sentence–external context via caches, hierarchical attention, and memory–augmented Transformers (Tu et al., 2018; Miculicich et al., 2018; Voita et al., 2018; Kim et al., 2019; Zhang et al., 2022). (Maruf et al., 2019) presents a comprehensive survey of DocMT. Beyond generic context, graph and discourse–aware methods encode structure explicitly: document graphs and coreference–aware encoders improve coherence (Ohtani et al., 2019; Xu et al., 2021). Large–scale document corpora and para–document resources continue to expand evaluation coverage (Ghussin et al., 2023).

LLM-based DocMT LLMs have recently demonstrated strong performance on MT tasks (Pang et al., 2025). Efforts to extend their capabilities to DocMT include fine-tuning on high-quality parallel data (Wu et al., 2024a; Ramos et al., 2025) and context-aware promoting (Wang et al., 2024a; Cui et al., 2024; Wang et al., 2023). More recently, growing attention has turned toward multi–agent DocMT pipelines which coordinates roles and memories to sustain translation consistency (Wu et al., 2024b, 2025; Wang et al., 2025). While effective, these designs can be token–intensive. Concurrently, Dutta et al. (2025) also employs LLMs to build document-level relation graphs, but unlike our approach, they model only the existence of links between chunks without explicitly defining relation types, and rely on external memory modules for context summarisation. In contrast, our work introduces discord-graph guided approach, offering structured, compact context that enhances cohesion and terminology accuracy with minimal overhead, aligning naturally with discourse theory (Mann and Thompson, 1988; Taboada and Mann, 2006).

5 Conclusions

We presented TRANSGRAPH, a discourse–guided DocMT framework that conditions chunk translations on a compact, relation–labeled graph context. Experiments on ACL 60/60, BWB, and GuoFeng show consistent improvements in document–level BLEU/COMET and substantial gains in terminology accuracy relative to sentence–level, single–pass, and agentic baselines, while reducing the total number of tokens and calls. LLM–assisted cohesion evaluation confirms benefits for coreference and conjunction handling. Our findings suggest that modest, well–structured discourse retrieval is a robust lever for document translation with LLMs.

Limitations

First, discourse graphs rely on LLM relation labeling. Although our analysis shows high accuracy and reasonable cross–lingual consistency, residual mislabels may inject noisy context. Second, while graph–retrieved context reduces token budgets relative to agentic pipelines, scalability still depends on chunking granularity and windowing. Very long documents with dense relation structure increase retrieval and prompt assembly costs. Moreover, LLMs remain sensitive to context placement and length (Dai et al., 2019; Beltagy et al., 2020; Liu et al., 2023). Future works could combine retrieval–augmented compression and learned relation selection to further improve robustness and efficiency.

Acknowledgements

The authors are grateful to the anonymous reviewers for their helpful comments. This research is supported by Monash eResearch capabilities, including M3.

References

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv*.

Menglong Cui, Jiangcun Du, Shaolin Zhu, and Deyi Xiong. 2024. Efficiently exploring large language models for document-level machine translation with in-context learning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 10885–10897, Bangkok, Thailand. Association for Computational Linguistics.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of ACL*, pages 2978–2988.

Himanshu Dutta, Sunny Manchanda, Prakhar Bapat, Meva Ram Gurjar, and Pushpak Bhattacharyya. 2025. Graft: A graph-based flow-aware agentic framework for document-level machine translation.

Yusser Al Ghussin, Jingyi Zhang, and Josef van Genabith. 2023. Exploring paracrawl for document-level neural machine translation.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. 2024. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Yuchen Eleanor Jiang, Tianyu Liu, Shuming Ma, Dongdong Zhang, Ryan Cotterell, and Mrinmaya Sachan. 2023a. Discourse centric evaluation of machine translation with a densely annotated parallel corpus. In *Proceedings of the 2023 Conference of the Association for Computational Linguistics: Human Language Technologies*, pages 1550–1565, Toronto, Canada. Association for Computational Linguistics.

Yuchen Eleanor Jiang, Tianyu Liu, Shuming Ma, Dongdong Zhang, Mrinmaya Sachan, and Ryan Cotterell. 2023b. Discourse-centric evaluation of document-level machine translation with a new densely annotated parallel corpus of novels. In *Proceedings of ACL (Long Papers)*, pages 7853–7872, Toronto, Canada.

Yuchen Eleanor Jiang, Tianyu Liu, Shuming Ma, Dongdong Zhang, Mrinmaya Sachan, and Ryan Cotterell. 2023c. Discourse centric evaluation of machine translation with a densely annotated parallel corpus. *Preprint*, arXiv:2305.11142.

Jiho Kim, Yeonsu Kwon, Yohan Jo, and Edward Choi. 2023. KG-GPT: A general framework for reasoning on knowledge graphs using large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9410–9421, Singapore. Association for Computational Linguistics.

Yunsu Kim, Kilian Happe, Pavel Petrushkov, Josef van Genabith, and Hermann Ney. 2019. When and why is document-level context useful in neural machine translation? In *Proceedings of the Fourth Workshop on Discourse in Machine Translation*, pages 24–34.

Zongyao Li, Zhiqiang Rao, Hengchao Shang, Jiaxin Guo, Shaojun Li, Daimeng Wei, and Hao Yang. 2025. Enhancing large language models for document-level translation post-editing using monolingual data. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 8830–8840, Abu Dhabi, UAE. Association for Computational Linguistics.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts. *Preprint*, arXiv:2307.03172.

Chenyang Lyu, Zefeng Du, Jitao Xu, Yitao Duan, Minghao Wu, Teresa Lynn, Alham Fikri Aji, Derek F. Wong, and Longyue Wang. 2024. A paradigm shift: The future of machine translation lies with large language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 1339–1352, Torino, Italia. ELRA and ICCL.

William C. Mann, Christian M. I. M. Matthiessen, and Sandra A. Thompson. 1989. Rhetorical structure theory and text analysis. Technical Report ISI/RR-89-242, Information Sciences Institute, University of Southern California, Marina del Rey, CA.

William C. Mann and Sandra A. Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text & Talk*, 8(3):243–281.

Sameen Maruf, Fahimeh Saleh, and Gholamreza Haffari. 2019. A survey on document-level neural machine translation: Methods and evaluation. *arXiv*.

Lesly Miculicich, Dhananjay Ram, Nikolaos Pappas, and James Henderson. 2018. Document-level neural machine translation with hierarchical attention networks. In *Proceedings of EMNLP*, pages 2947–2954.

Takumi Ohtani, Hidetaka Kamigaito, Masaaki Nagata, and Manabu Okumura. 2019. Context-aware neural machine translation with coreference information. In *Proceedings of the Fourth Workshop on Discourse in Machine Translation (DiscoMT 2019)*, pages 45–50, Hong Kong, China. Association for Computational Linguistics.

Jianhui Pang, Fanghua Ye, Derek Fai Wong, Dian Yu, Shuming Shi, Zhaopeng Tu, and Longyue Wang. 2025. Salute the classic: Revisiting challenges of machine translation in the age of large language models. *Transactions of the Association for Computational Linguistics*, 13:73–95.

Libo Qin, Zecheng Ji, Wanqi Wang, Wanxiang Che, and Ting Liu. 2022. CoDoNMT: Modeling cohesion devices for document-level neural machine translation. In *Proceedings of COLING*, pages 5232–5245.

Miguel Moura Ramos, Patrick Fernandes, Sweta Agrawal, and Andre Martins. 2025. Multilingual contextualization of large language models for document-level machine translation. In *Second Conference on Language Modeling*.

Elizabeth Salesky, Kareem Darwish, Mohamed Al-Badrashiny, Mona Diab, and Jan Niehues. 2023. Evaluating multilingual speech translation under realistic conditions with resegmentation and terminology. In *Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023)*, pages 62–78, Toronto, Canada. Association for Computational Linguistics.

Maite Taboada and William C. Mann. 2006. Rhetorical structure theory: Looking back and moving ahead. *Discourse Studies*, 8(3):423–459.

Xin Tan, Longyin Zhang, and Guodong Zhou. 2022. Discourse cohesion evaluation for document-level neural machine translation. *Preprint*, arXiv:2208.09118.

Zhaopeng Tu, Yang Liu, Shuming Liu, and Maosong Zhou. 2018. Learning to remember translation history with a continuous cache. In *Transactions of the Association for Computational Linguistics*, volume 6, pages 407–420.

Giorgos Vernikos, Brian Thompson, Prashant Mathur, and Marcello Federico. 2022. Embarrassingly easy document-level MT metrics: How to convert any pretrained metric into a document-level metric. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 118–128, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Elena Voita, Pavel Serdyukov, Rico Sennrich, and Ivan Titov. 2018. Context-aware neural machine translation learns anaphora resolution. In *Proceedings of ACL*, pages 1264–1274.

Longyue Wang, Zefeng Du, Wenxiang Jiao, Chenyang Lyu, Jianhui Pang, Leyang Cui, Kaiqiang Song, Derek Wong, Shuming Shi, and Zhaopeng Tu. 2024a. Benchmarking and improving long-text translation with large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 7175–7187, Bangkok, Thailand. Association for Computational Linguistics.

Longyue Wang, Siyou Liu, Minghao Wu, Wenxiang Jiao, Xing Wang, Jiahao Xu, Zhaopeng Tu, Liting Zhou, Yan Gu, Weiyu Chen, Philipp Koehn, Andy Way, and Yulin Yuan. 2024b. Findings of the wmt 2024 shared task on discourse-level literary translation. In *Proceedings of the Ninth Conference on Machine Translation*.

Longyue Wang, Chenyang Lyu, Tianbo Ji, Zhirui Zhang, Dian Yu, Shuming Shi, and Zhaopeng Tu. 2023. Document-level machine translation with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16646–16661, Singapore. Association for Computational Linguistics.

Yutong Wang, Jiali Zeng, Xuebo Liu, Derek F. Wong, Fandong Meng, Jie Zhou, and Min Zhang. 2025. DelTA: An online document-level translation agent based on multi-level memory. In *The Thirteenth International Conference on Learning Representations*.

Minghao Wu, Thuy-Trang Vu, Lizhen Qu, George Foster, and Gholamreza Haffari. 2024a. Adapting large language models for document-level machine translation. *arXiv preprint arXiv:2401.06468*.

Minghao Wu, Jiahao Xu, Yulin Yuan, Gholamreza Haffari, Longyue Wan, Weihua Luo, and Kaifu Zhang. 2025. (perhaps) beyond human translation: Harnessing multi-agent collaboration for translating ultra-long literary texts. *Transactions of the Association for Computational Linguistics*, 13:901–922.

Minghao Wu, Jiahao Xu, Yulin Yuan, Gholamreza Haffari, Longyue Wang, Weihua Luo, and Kaifu Zhang. 2024b. Transagents: Build your translation company with language agents. In *Proceedings of EMNLP 2024: System Demonstrations*.

Haoran Xu, Young Jin Kim, Amr Sharaf, and Hany Hassan Awadalla. 2024. A paradigm shift in machine translation: Boosting translation performance of large language models. In *The Twelfth International Conference on Learning Representations*.

Mingzhou Xu, Liangyou Li, Derek. F. Wong, Qun Liu, and Lidia S. Chao. 2021. Document graph for neural machine translation.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Shaolei Zhang, Yang Feng, and 1 others. 2022. Rethinking document-level neural machine translation. In *Findings of ACL*, pages 3537–3548.

A Prompt Templates

We provide the prompt templates for TRANSGRAPH in this section. Listing 1 shows the prompt template for document chunking, Listing 2 shows the prompt template for relation labeling and finally, Listing 3 is the prompt we used for iterative chunk translation.

Prompt templates used for evaluating the cohesion are also provided, including the annotation (Listing 4, Listing 5) and evaluation (Listing 6, Listing 7) on coreferences and conjunctions, respectively.

B Examples

We provide the example of document chunking performed by Qwen3-32B in Listing 8, while the examples for the relation identification process are given in Table 10.

```
You are a document chunker. Here is an (incomplete) part of a long document:
{chunk_content}

Split the given text into contiguous, sentence-preserving chunks with coherent, adjacent sentences
only. Remember to: (i) never split a sentence, (ii) keep topically related, adjacent sentences
together, and (iii) if the last sentence is incomplete, do NOT output it - mark it as carry_over.

Return ONLY valid JSON with this schema:
{
  "chunks": [
    {
      "chunk_id": "<int|string>",
      "rationale": "<2 short sentences on why you choose the following sentences for this chunk>"
      "sentence_indices": [<ints, in order>]
      "carry_over": <true|false>,
    }
  ],...
}
```

Listing 1: Document chunking prompt template.

```
You are a text relation analyzer. Your task is to examine two chunks of text, taken from the same
document, **Chunk {i}** and **Chunk {j}** and determine if they share any meaningful connection. Only
 define a relation when there truly is one; if the two chunks are topically or semantically
independent, indicate that explicitly.

### Instructions

1. **Identify a Relation (if any)**
   - Read **Chunk {j}** (the later chunk) and **Chunk {i}** (an earlier chunk).
   - Ask yourself:
     - Do they refer to the same entities, events, or ideas?
     - Does Chunk {j} expand, clarify, contrast, or provide background for something in Chunk {i}?
     - Does one chunk continue a thought begun in the other (e.g., cause -> effect, premise ->
conclusion)?
   - If there is an explicit connection, choose the most precise descriptor from the following
categories:
     - **Background->Core** ("Chunk {i} provides necessary background for Chunk {j}")
     - **Core->Detail** ("Chunk {j} drills down into a specific subpoint that Chunk {i} introduced")
     - **Motivation->Method or Problem->Solution** ("Chunk {i} states a problem; Chunk {j} describes
the solution.")
     - **Cause->Effect** (e.g., "Chunk {i} describes an experiment; Chunk {j} shows the resulting
performance drop.")
     - **Contrast** (e.g., "Chunk {j} explicitly contrasts with Chunk {i}'s claim.")
     - **Comparison** (e.g., "Both chunks compare two architectures.")
     - **Condition** (e.g., "Chunk {i} says 'if X, then Y,' and Chunk {j} describes what happens
under that condition.")
     - **Evaluation** (e.g., "Chunk {j} evaluates the approach introduced in Chunk {i}.")
     - **Entity Coreference** ("Both chunks mention the same dataset, model, or variable.")
     - **Terminology Definition** ("Chunk {i} defines a term that appears in Chunk {j}.")

2. **Output Format**
   - Output strictly a dictionary with keys: reason, relation, direction. Reason is a short
explanation for the relation. If the relation is 'none', reason should be 'no relation found'.
Direction is the direction of the relation, which can be 'forward' or 'backward'.
   - **Do not** include any extra commentary, annotations, or keys. Only the dictionary should be
returned.

# Chunk {i}:
{chunk_i}
# Chunk {j}:
{chunk_j}

Your response:
```

Listing 2: Relation identification prompt template.

```
You are a high-quality translation assistant. Your task is to translate one specific chunk of the
source document from {src_lang} into {tgt_lang}, using the following context to ensure consistent
terminology, style, and meaning.

A. RELATED CHUNKS (and WHY THEY MATTER)

Below are all chunks that share a meaningful connection with the "Current Chunk."
For each related chunk, you have:
  - Its "Chunk ID" (an integer).
  - The full source-language text of that chunk.
  - The detected relation-type between that chunk and the current one.
  - A brief "Reason" sentence explaining why this relation helps guide terminology or meaning in the
current chunk.

Use these related-chunk definitions to preserve consistent translations of any key terms or ideas
that overlap.

{related_chunks}

B. CURRENT CHUNK TO TRANSLATE

Chunk ID: {chunk_id}

Source Text:
{chunk_text}

C. SPECIFIC INSTRUCTIONS

1. **Consistent Terminology:**
   - If any key term has been introduced or defined in a Related Chunk, use **exactly the same target
-language rendering** here.
   - If a Related Chunk indicates an acronym expansion, be sure to translate the acronym and its
expanded form consistently (following how it was rendered earlier).

2. **Preserve Coreference & Referential Integrity:**
   - If the Current Chunk refers back to an entity or concept defined earlier, ensure you use the
same translation for that concept-exactly as in the Related Chunk's translation.

3. **Translate Only the Current Chunk:**
   - Do not attempt to retranslate the entire document or other chunks.
   - Your output should be **only** the translated text of {chunk_id}.
   - Do **not** include any commentary, footnotes, or explanations-just the final translation block.

4. **Formatting:**
   - Keep paragraph breaks as in the source.
   - If the source chunk has multiple paragraphs, translate each paragraph and preserve line breaks.
   - If the source chunk contains inline code, variable names, or labels, keep them in-English or
code-style without adding extra formatting.


D. OUTPUT

Provide **only** the translated text of Chunk {chunk_id} in {tgt_lang}, respecting all the
instructions above.
```

Listing 3: Iterative chunk translation prompt template.

```
# Source Pronoun Annotation
{...} Role definition is omitted due to space limitations
## Task Definition
**Pronouns** are words that refer to entities mentioned elsewhere in the text or understood from
context, including:
- **Personal pronouns**: I, you, he, she, it, we, they, me, him, her, us, them
- **Possessive pronouns**: my, your, his, her, its, our, their, mine, yours, hers, ours, theirs
- **Demonstrative pronouns**: this, that, these, those (when used as pronouns, not determiners)
- **Reflexive pronouns**: myself, yourself, himself, herself, itself, ourselves, yourselves,
themselves
- **Relative pronouns**: who, whom, whose, which, that (when introducing relative clauses)

## Annotation Instructions

### Step 1: Read the entire document
Understand the content, identify all entities, and track referential relationships throughout the
text.

### Step 2: Identify ALL pronouns
Scan systematically through the document for every pronoun that refers to a specific entity or
concept.

### Step 3: Determine pronoun type and referent
- Classify the grammatical type of each pronoun
- Identify what specific entity or concept each pronoun refers to

### Step 4: Apply inline annotation
First copy each pronoun exactly as it appears in the original text, then add the attributes after it.

## Annotation Format
Use this exact format:
‘‘‘
[pronoun]<type="[pronoun_type]” referent="[what_it_refers_to]">
‘‘‘

**Attribute specifications:**
- ‘type‘: One of: ‘personal‘, ‘possessive‘, ‘demonstrative‘, ‘reflexive‘, ‘relative‘
- ‘referent‘: The specific noun phrase or concept that this pronoun refers to

## Pronoun Type Guidelines

- **personal**: I, you, he, she, it, we, they, me, him, her, us, them
- **possessive**: my, your, his, her, its, our, their, mine, yours, hers, ours, theirs
{...} more types are omitted due to space limitations


## Examples

### Example 1 (Basic Pronouns):

**Source document**:
‘‘‘
John bought a new car yesterday. He drove it to work this morning. Mary saw him and thought the car
was beautiful. She told him that she liked it very much.
‘‘‘

**Expected Output**:
‘‘‘
John bought a new car yesterday. [He]<type="personal” referent="John”> drove [it]<type="personal”
referent="a new car”> to work this morning. Mary saw [him]<type="personal” referent="John”> and
thought the car was beautiful. [She]<type="personal” referent="Mary”> told [him]<type="personal”
referent="John”> that [she]<type="personal” referent="Mary”> liked [it]<type="personal” referent="the
 car”> very much.
‘‘‘
{...} more examples are omitted due to space limitations
```

Listing 4: Prompt template for instructing LLM to annotate the reference words in the source text

```
# Source Conjunction Annotation
{...} Role definition is omitted due to space limitations
## Task Definition
**Conjunctive expressions** are words or phrases that signal logical relationships between clauses or
 sentences, including:
- **Coordinating conjunctions**: and, but, or, nor, for, so, yet
- **Subordinating conjunctions**: because, since, although, while, if, when, before, after, unless,
etc.
- **Conjunctive adverbs**: however, therefore, furthermore, meanwhile, consequently, nevertheless,
moreover, etc.
- **Transitional phrases**: in addition, on the other hand, as a result, for example, in contrast,
etc.
- **Correlative conjunctions**: both...and, either...or, not only...but also, etc.

## Annotation Instructions
### Step 1: Read the entire document
Understand the content and identify the logical flow and relationships between clauses and sentences.

### Step 2: Identify ALL conjunctive expressions
Scan systematically through the document for every word or phrase that connects ideas or shows
logical relationships.

### Step 3: Determine conjunction type and logical relationship
- Classify the grammatical type of each conjunctive expression
- Identify what type of logical connection it signals

### Step 4: Apply inline annotation
First copy each conjunctive expression exactly as it appears in the original text, then add the
attributes after it.

## Annotation Format
Use this exact format:
‘‘‘
[conjunction]<type="[conjunction_type]” relationship="[logical_relationship]">
‘‘‘

**Attribute specifications:**
- ‘type‘: One of: ‘coordinating‘, ‘subordinating‘, ‘conjunctive_adverb‘, ‘transitional_phrase‘, ‘
correlative‘
- ‘relationship‘: One of the logical relationship categories listed below

## Logical Relationship Categories
- **addition**: adding information (and, furthermore, moreover, in addition)
{...} more categories are omitted due to space limitations

## Conjunction Type Guidelines

- **coordinating**: and, but, or, nor, for, so, yet
- **subordinating**: because, since, although, while, if, when, before, after, unless, though,
whereas, etc.
{...} more types are omitted due to space limitations

## Examples

### Example 1 (Basic Conjunctions):

**Source document**:
‘‘‘
John was tired, but he had to continue working. Therefore, he decided to have a cup of coffee. He
drank it quickly and felt more energetic.
‘‘‘
**Expected Output**:
‘‘‘
John was tired, [but]<type=”coordinating” relationship=”contrast”> he had to continue working. [
Therefore]<type=”conjunctive_adverb” relationship=”result”>, he decided to have a cup of coffee. He
drank it quickly [and]<type=”coordinating” relationship=”addition”> felt more energetic.
‘‘‘
{...} more examples are omitted due to space limitations
```

Listing 5: Prompt template for conjunction annotation

```
# Reference Cohesion Translation Quality Evaluation
{...} Role definition is omitted due to space limitations
## Task Overview

You will receive an English source text that has been pre-annotated with pronoun information, along
with a translation in a target language. Your job is to evaluate each annotated pronoun by
determining:
1. How it was translated in the target language
2. Whether the translation is correct
3. If incorrect, what type of error occurred

## Evaluation Guidelines

For each annotated pronoun in the source text, you must add three new attributes to the existing
annotation:

### Required Attributes to Add:
- **target_translation**: How the pronoun was rendered in the target language
- **is_correct**: Whether the translation is accurate (true/false)
- **error_type**: Type of error if translation is incorrect (null if correct)

### target_translation Values:
1. **Specific translation word(s)**: The actual translated pronoun (e.g., ”Er”, ”elle”)
2. **”omitted”**: The pronoun was appropriately omitted (common in pro-drop languages like Chinese/
Japanese)
3. **”missing”**: The pronoun should have been translated but is absent

### is_correct Logic:
- If ‘target_translation‘ = specific word(s) -> ‘is_correct‘ can be true or false
- If ‘target_translation‘ = "omitted" -> ‘is_correct‘ must be true (appropriate omission)
- If ‘target_translation‘ = "missing" -> ‘is_correct‘ must be false (inappropriate absence)

### error_type Categories:
- **”null”**: No error (translation is correct)
- **”gender_mismatch”**: Wrong gender (he->she, him->her, etc.)
{...} more types are omitted due to space limitations

## Output Format

Return the complete annotated source text with the three new attributes added to each pronoun
annotation:
‘‘‘
[pronoun]<type=”...” referent=”...” target_translation=”...” is_correct=”true|false” error_type
=”...”>
‘‘‘

## Examples

### English Source Text
Tom and his sister went to the park. She found a ball and he picked it up. They decided to play
together.
{...} English annotation is omitted due to space limitations
### German Translation
**Translation**: Tom und seine Schwester gingen in den Park. Er fand einen Ball und er hob ihn auf.
waren glucklich zusammen zu spielen.

**Expected Output**:
‘‘‘
Tom and [his]<type=”possessive” referent=”Tom” target_translation=”seine” is_correct=”true”
error_type=”null”> sister went to the park. [She]<type=”personal” referent=”his sister”
target_translation=”Er” is_correct=”false” error_type=”wrong_referent”> found a ball and [he]<type=”
personal” referent=”Tom” target_translation=”er” is_correct=”true” error_type=”null”> picked [it]<
type=”personal” referent=”a ball” target_translation=”ihn” is_correct=”true” error_type=”null”> up. [
They]<type=”personal” referent=”Tom and his sister” target_translation=”missing” is_correct=”false”
error_type=”missing_translation”> decided to play together.
‘‘‘
{...} More examples are omitted due to space limitations
```

Listing 6: Prompt template for instructing LLM to evaluate the reference cohensiveness

```
# Conjunction Cohesion Translation Quality Evaluation
{...} Role definition is omitted due to space limitations
## Task Overview
You will receive an English source text that has been pre-annotated with conjunction information,
along with a translation in a target language. Your job is to evaluate each annotated conjunction by
determining:
1. How it was translated in the target language
2. Whether the translation preserves the correct logical relationship
3. If incorrect, what type of error occurred

## Evaluation Guidelines
For each annotated conjunction in the source text, you must add three new attributes to the existing
annotation:

### Required Attributes to Add:
- **target_translation**: How the conjunction was rendered in the target language
- **is_correct**: Whether the translation preserves the logical relationship (true/false)
- **error_type**: Type of error if translation is incorrect (null if correct)

### target_translation Values:
1. **Specific translation word(s)**: The actual translated conjunction (e.g., ”aber”, ”mais”)
2. **”missing”**: The conjunction should have been translated but is absent

### is_correct Logic:
- If ‘target_translation‘ = specific word(s) -> ‘is_correct‘ can be true or false depending on
logical relationship
- If ‘target_translation‘ = ”missing” -> ‘is_correct‘ must be false (conjunction information lost)

### error_type Categories:
- **”null”**: No error (translation preserves correct logical relationship)
- **”wrong_conjunction”**: Conjunction translated but expresses wrong logical relationship
- **”missing_conjunction”**: Required conjunction is completely absent
- **”redundant_conjunction”**: Multiple conjunctions expressing same logical relationship
- **”inappropriate_addition”**: Adding conjunctions that create wrong logical relationships
- **”wrong_position”**: Conjunction in wrong syntactic position affecting meaning

## Output Format
Return the complete annotated source text with the three new attributes added to each conjunction
annotation:
‘‘‘
[conjunction]<type=”...” relationship=”...” target_translation=”...” is_correct=”true|false”
error_type=”...”>
‘‘‘

## Examples
### German Translation
**Annotated Source**:
{...} Source annotation is omitted due to space limitations
**Target Translation**: Das Wetter war schlecht, so entschieden wir uns zu wandern. Zuerst packten
wir unsere Taschen. Dann verlieben wir fruh, weil wir den Verkehr vermeiden wollten. Obwohl es zu
regnen begann, obwohl wir unsere Reise fortsetzten.

**Expected Output**:
‘‘‘
The weather was bad, [but]<type=”coordinating” relationship=”contrast” target_translation=”so”
is_correct=”false” error_type=”wrong_conjunction”> we decided to go hiking. [First]<type=”
conjunctive_adverb” relationship=”sequence” target_translation=”Zuerst” is_correct=”true” error_type
=”null”>, we packed our bags. [Then]<type=”conjunctive_adverb” relationship=”sequence”
target_translation=”Dann” is_correct=”true” error_type=”null”> we left early [because]<type=”
subordinating” relationship=”cause” target_translation=”weil” is_correct=”true” error_type=”null”> we
 wanted to avoid traffic. [Although]<type=”subordinating” relationship=”concession”
target_translation=”Obwohl, obwohl” is_correct=”false” error_type=”redundant_conjunction”> it started
 raining, we continued our journey.
‘‘‘
{...} More examples are omitted due to space limitations
```

Listing 7: Prompt template for instructing LLM to evaluate the conjunction cohensiveness

```
Chunk 1
Hello. My name is Asaf Harari. And I will present our paper, Few-Shot Tabular Data Enrichment Using
Fine-Tuned Transformers Architectures. Data scientists analyze data and mainly focus on the
manipulating the data’s existing features. But sometimes, these features are limited. Feature
generation using another data source may add substantial information. Our research goal is automatic
tabular data enrichment using external sources’ free text.

Chunk 2
Assume we have a tabular dataset and a knowledge base. We need an automatic process which involves
entity linking and text analysis to extract new features from the knowledge base’s free text. Our
framework FeSTE is exactly this automatic process. So let’s see an example in a dataset fed into
FeSTE. In this example, the dataset is university dataset. When its goal is to classify universities
into low ranking universities and high-ranking universities. As knowledge base, we use Wikipedia.

Chunk 3
The first phase of FeSTE is entity linking. When each entity, in this example the university name, is
 linked to an entity within the knowledge base. And and the text of the entities of the knowledge
base is extracted and added to the dataset. In this example, the text is the Wikipedia page’s
abstract. Now, we need to generate or extract features from the retrieved text. So, we need to ah
feature extraction phase ah which includes text analysis. And this is the main novelty of this paper
and I will deep dive into it in the next slides.

Chunk 4
After the feature extraction phase, there is a feature generation phase when we use the extracted
features to generate a small number of new features. First generate ah features in the number of
classes of the original dataset. In this example, the original dataset has two classes. So, FeSTE
generates two new features. But if the dataset has five classes, FeSTE generates five new features.
Each feature represents the likelihood for each class. To analyze the text, we use the current state-
of-the-art of text analysis, which are transformer based language models as BERT, GPT, XLNet and etc.

Chunk 5
It is but it is not likely that we can train language models using the input datasets. So a naive
approach will be ah target task finetuning. So, in the feature extraction phase, we can download
pretrained language models, finetune the language model over the target dataset. In this example to
finetune the language model, to classify ah to classify text into classes, abstract into classes, low
 or high. Receive the language model output, which is the likelihood for each class and use as new
features. The problem with this approach is datasets may have few distinct entities / texts. In our
experiment, almost half of the datasets contain less than four hundred samples and the smallest
dataset contain thirty five samples in its, in a training set. So to finetune a language model over
ah this dataset will be ineffective.

...

Chunk 11
Here are the results for our experiments. You can see that we compare our our framework to target
dataset finetuning, target task finetuning, and a MTDNN preliminary finetuning. And our reformulated
finetuning achieves the best result, the best performance. While MTDNN achieved two percent
improvement over the target dataset finetuning. Our approach achieved six percent improvement. When
we look on the small ah dataset, we can see that the performance of MTDNN decreases and the
improvement of the prelim, the preliminary multitask finetuning phase decreases to one point five
percent. But our performance increased to eleven percent compared to the target task finetuning alone
.

Chunk 12
For summing, FeSTE enables few shot enrichment from thirty five samples in our experiments. It uses
one architecture for all tasks and datasets. And it keeps the head of ah of the model. But it adds
reformulation phase. It augments the train set and it needs a target value with semantic meaning so
we can feed it into the language model and use it in the sentence pair classification problem. Thank
you.
```

Listing 8: Example of document chunking.

Relation	**Chunk i**	**Chunk j**
Terminology Definition	Hello. My name is Asaf Harari. And I will present our paper, **Few-Shot Tabular Data Enrichment Using Fine-Tuned Transformers Architectures.**	So let's see an example in a dataset fed into **FeSTE**. In this example, the dataset is university dataset. When its goal is to classify universities into low ranking universities and high-ranking universities. As knowledge base, we use Wikipedia.
Entity Coreference	So let's see an example in a dataset fed into **FeSTE**. In this example, the dataset is university dataset. When its goal is to classify universities into low ranking universities and high-ranking universities. As knowledge base, we use Wikipedia.	The first phase of **FeSTE** is entity linking. When each entity, in this example the university name, is linked to an entity within the knowledge base. And and the text of the entities of the knowledge base is extracted and added to the dataset. In this example, the text is the Wikipedia page's abstract.
Background → Core	Data scientists analyze data and mainly focus on the manipulating the data's existing features. But sometimes, these features are limited. Feature generation using another data source may add substantial information. **Our research goal is automatic tabular data enrichment using external sources' free text.**	Assume we have a tabular dataset and a knowledge base. We need an automatic process which involves entity linking and text analysis to extract new features from the knowledge base's free text. **Our framework FeSTE is exactly this automatic process.**
Core → Detail	Assume we have a tabular dataset and a knowledge base. We need an automatic process which involves entity linking and text analysis to extract new features from the knowledge base's free text. Our framework FeSTE is exactly this automatic process.	**So let's see an example in a dataset fed into FeSTE.** In this example, the dataset is university dataset. When its goal is to classify universities into low ranking universities and high-ranking universities. As knowledge base, we use Wikipedia.
Contrast	The state-of-the-art in multitask ah multitask finetuning called MTDNN. **In MTDNN, MTDNN maintains ah heads in the number of tasks in the training set**. So, in this example there are four tasks in the training set, so MTDNN maintain four heads as you can see at the image. And it samples a random batch from ah from the training set. And if they random batch belongs to a, for example single sentence classification task, it executes forward and backward paths through the first head. And if the random batch belongs to pairwise ranking task, it executes forward and backward path through the last head.	In our scenario, ah tabular datasets vary in the number of classes. So there are many tasks. MTDNN maintained number of classes, heads, output layers. And the additional, additionally MTDNN needs to initialize new heads for a new dataset with a new task. **Our approach, called task reformulation finetuning is, in our approach task reformulation finetuning, instead of maintaining multiple heads, we reformulate each dataset into a sentence per classification problem, which is two classes' tasks.**
Comparison	**The state-of-the-art in multitask ah multitask finetuning called MTDNN. In MTDNN, MTDNN maintains ah heads in the number of tasks in the training set**. So, in this example there are four tasks in the training set, so MTDNN maintain four heads as you can see at the image. And it samples a random batch from ah from the training set. And if they random batch belongs to a, for example single sentence classification task, it executes forward and backward paths through the first head. And if the random batch belongs to pairwise ranking task, it executes forward and backward path through the last head.	Here are the results for our experiments. **You can see that we compare our our framework to target dataset finetuning, target task finetuning, and a MTDNN preliminary finetuning. And our reformulated finetuning achieves the best result, the best performance. While MTDNN achieved two percent improvement over the target dataset finetuning**. Our approach achieved six percent improvement. When we look on the small ah dataset, we can see that the performance of MTDNN decreases and the improvement of the prelim, the preliminary multitask finetuning phase decreases to one point five percent. But our performance increased to eleven percent compared to the target task finetuning alone.

Table 10: Relations between chunk pairs with blue-highlighted bold evidence provided by Qwen3-32B.