DoCIA: An Online Document-Level Context Incorporation Agent for Speech Translation

Xinglin Lyu♠, Wei Tang♠, Yuang Li♠, Xiaofeng Zhao♠, Ming Zhu♠, Junhui Li♣, Yunfei Lu♦, Min Zhang♠, Daimeng Wei♠, Hao Yang♠*, Min Zhang♣

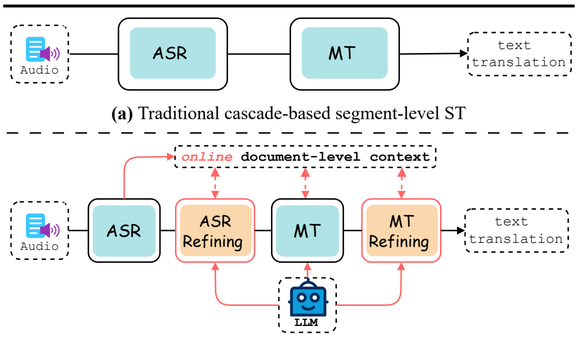
♠Huawei Translation Services Center, Beijing, China
♦Huawei Consumer Business Group, Beijing, China
♣School of Computer Science and Technology, Soochow University, Suzhou, China
`{lvxinglin1,tangwei133,zhangming186,yanghao30}@huawei.com`
`{lijunhui,minzhang}@suda.edu.cn`

Abstract

Document-level context is crucial for handling discourse challenges in text-to-text document-level machine translation (MT). Despite the increased discourse challenges introduced by noise from automatic speech recognition (ASR), the integration of document-level context in speech translation (ST) remains insufficiently explored. In this paper, we develop DoCIA, an online framework that enhances ST performance by incorporating document-level context. DoCIA decomposes the ST pipeline into four stages. Document-level context is integrated into the ASR refinement, MT, and MT refinement stages through auxiliary LLM (large language model)-based modules. Furthermore, DoCIA leverages document-level information in a multi-level manner while minimizing computational overhead. Additionally, a simple yet effective determination mechanism is introduced to prevent hallucinations from excessive refinement, ensuring the reliability of the final results. Experimental results show that DoCIA significantly outperforms traditional ST baselines in both sentence and discourse metrics across four LLMs, demonstrating its effectiveness in improving ST performance.[1]

1 Introduction

Speech translation (ST) involves translating spoken language into written text in a different language. Despite significant progress in recent years (Zhang et al., 2019a; Sperber and Paulik, 2020; Ye et al., 2021; Fang et al., 2022; Lei et al., 2023), incorporating document-level context into ST remains a major challenge due to the cross-modal nature of the task. This paper shifts the focus to document-level context[2] and examines how it can enhance machine translation (MT) when combined with automatic speech recognition (ASR) in the cascaded ST systems.



(a) Traditional cascade-based segment-level ST

(b) Proposed LLM-based and context-aware DoCIA for ST

Figure 1: The traditional cascade-based ST system (*top*) and our proposed DoCIA for ST (*bottom*). Differently, DoCIA introduces two refinement stages and is LLM-based and context-aware when translating i-th audio segment in a speech.

In a traditional cascaded ST system, as shown in Figure 1 (a), the ASR and MT models operate independently at the segment level. This leads to significant discourse-level issues due to the absence of inter-sentence context. These challenges become even more pronounced in ST, where ASR errors—such as misrecognizing entity pronouns or handling disfluencies—further complicate the translation process. Incorporating document-level context offers two key advantages: first, it can potentially correct ASR transcription errors by providing a broader understanding of the context; second, when integrated into the MT model, it helps address discourse phenomena such as entity inconsistencies, coreference resolution, and long-range dependencies (Sennrich et al., 2016; Zhang et al., 2018; Bao et al., 2021, 2023; Lyu et al., 2024). To fully leverage document-level context, we introduce DoCIA—**Do**cument-level **C**ontext **I**ncorporation **A**gent—an online framework specifically designed to improve ST performance by incorporating document-level context.

*Corresponding author: Hao Yang.

[1]Code is available at `https://github.com/xllyu-nlp/DoCIA`

[2]Also referred to as inter-segment context in ST.

Findings of the Association for Computational Linguistics: ACL 2025, pages 14910–14924
July 27 - August 1, 2025

End-to-end ST systems, which directly translate source-language speech into target-language text, can reduce the propagation of ASR errors. However, these systems suffer from limited interpretability and the challenge of scarce parallel ST data, making it expensive to develop a reliable and effective end-to-end solution. In contrast, the cascading approach—especially with the emergence of powerful large language models (LLMs) (OpenAI, 2023; Google, 2023; Dubey et al., 2024)—provides a more efficient and flexible alternative. The cascading model enables modular optimization in ST, allowing LLMs to be used to enhance performance at various stages of the process. As shown in Figure 1 (b), DoCIA takes full advantage of the scalability and flexibility inherent in the cascading approach by breaking the ST process into four key stages: ASR, ASR refinement, MT, and MT refinement. Document-level context is incorporated during the latter three stages (ASR refinement, MT, and MT refinement), improving both transcription and translation through LLM-based agents. Crucially, the document-level context is updated *online* as each segment is processed, ensuring that the context remains current and relevant throughout the translation process.

In addition, we employ two techniques — *a multi-level context integration strategy* and *a refinement determination mechanism* — to enhance the performance of DoCIA. First, while document-level context can be beneficial, it often includes redundant information, with only a small portion being relevant to discourse issues (Kang et al., 2020). Using all available context indiscriminately can even degrade ST performance and increase computational overhead. To address this, we propose a multi-level context integration strategy that retains the advantages of document-level context while reducing redundancy. Second, our two refinement stages are designed to resolve inter-segment inconsistencies using document-level context. In most cases, minimal adjustments are sufficient to address discourse-related issues, as extensive changes may introduce errors such as hallucinations or semantic distortions. To minimize these risks, we introduce a determination mechanism that ensures the refined text remains consistent with the original semantics, improving the output without introducing undesirable changes.

Overall, the main contributions of this paper are summarized as follows:

- We extend cascaded ST to four stages and introduce DoCIA, an online agent that enhances ST by progressively incorporating document-level context at each text-to-text stage.

- We propose two techniques to enhance DoCIA: a multi-level document context integration strategy that selectively incorporates context, and a simple determination mechanism to prevent hallucinations during refinement.

- We validate DoCIA across five ST directions using four LLMs, including both closed- and open-source models, highlighting the importance of document-level context in ST.

2 DoCIA: Document-level Context Incorporation Agent

We propose DoCIA, an online agent designed to enhance speech translation (ST) by effectively leveraging document-level context. DoCIA operates in a cascaded four-stage process: ASR, ASR refinement, translation, and translation refinement. Document-level context is incorporated during the ASR refinement, translation, and translation refinement stages (Section 2.1). To optimize context utilization, we introduce a multi-level integration strategy, splitting the context into short- and long-memory components (Section 2.2). To prevent hallucinations during refinement, we also propose an effective determination mechanism (Section 2.3).

2.1 Overview of DoCIA

Given a speech $\mathcal{A} = \{a_1, a_2, \cdots, a_N\}$, which consists of N audio segments, DoCIA translates these segments sequentially. The overview of DoCIA is illustrated in Figure 2. To explain the translation process, let us consider the i-th audio segment a_i, as an example. The translation process in DoCIA involves four key stages, which produce the following outputs for a_i : the draft ASR result $\bar{s}_i$, the refined ASR result s_i, the draft translation $\bar{t}_i$, and the final refined translation t_i.

ASR Stage. First, DoCIA generates the draft transcription $\bar{s}_i$ of a_i using an ASR model:

$$\bar{s}_i = \mathrm{ASR}(a_i). \tag{1}$$

Here, ASR refers to the ASR model. Note that in this stage we obtain draft transcription at the segment level.

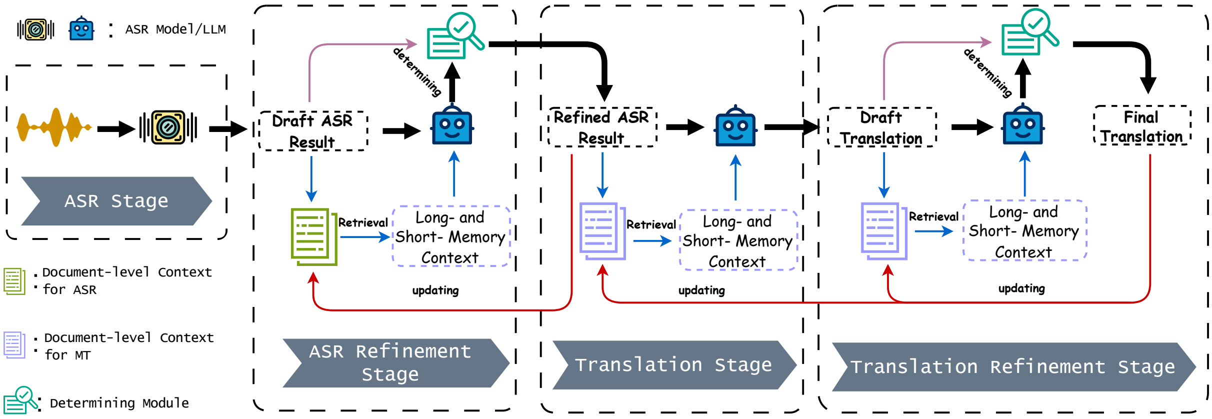


Figure 2: The overall illustration of DoCIA when translating i-th audio segment in a speech. The blue, purple and red lines denote the **context retrieving**, **refinement determining** and **context updating** processes, respectively.

ASR Refinement Stage. In this stage, DoCIA aims to correct errors in the draft transcription $\bar{s}_i$ and enhance its quality by incorporating document-level ASR context, denoted as $\mathcal{C}_{asr} = (s_1, \cdots, s_{i-1})$. DoCIA uses an LLM to obtain the refined transcription s_i via:

$$s_i = \text{argmax } p(s_i|\bar{s}_i, inst_{ar}, C_{asr}, \theta_{llm}), \quad (2)$$

where $inst_{ar}$ represents the instruction for the context-aware ASR refinement task.[3] $C_{asr} \subseteq \mathcal{C}_{asr}$ is the selected document-level context, which is determined using the strategy described in Section 2.2. The parameter θ_{llm} refers to the parameters of the LLM.

Translation Stage. In this stage, DoCIA similarly uses the LLM to translate the transcription s_i while incorporating both the source-side and target-side document-level context $\mathcal{C}_{asr}$ and $\mathcal{T}_{tr} = (t_1, \cdots, t_{i-1})$, resulting in the draft translation $\bar{t}_i$ for the audio segment a_i. This process is expressed as follows:

$$\bar{t}_i = \text{argmax } p(\bar{t}_i|s_i, inst_{mt}, C_{tr}, \theta_{llm}), \quad (3)$$

where $inst_{mt}$ represents the instruction for the context-aware translation task. The document-level context for translation, C_{tr}, combines the source-side context C_{asr} and the target-side context T_{tr} (also referred to as inter-segment translation history). T_{tr} contains the corresponding refined translations of the segments in C_{asr}.

Translation Refinement Stage. In this stage, we further leverage document-level context to improve the translation through a translation refinement process. Unlike the initial translation stage, where the focus is on generating a translation, the goal here is to enhance the word choice in the draft translation and ensure better cohesion and coherence with the preceding translation history. This process mimics the correction preferences typically applied by human translators. DoCIA again uses the LLM to perform the refinement. Given the draft translation $\bar{t}_i$ for the i-th audio segment a_i, DoCIA refines it by incorporating document-level context. The refinement process is expressed as follows:

$$t_i = \text{argmax } p(t_i|s_i, inst_{tr}, \bar{t}_i, C_{tr}, \theta_{llm}), \quad (4)$$

where $inst_{tr}$ denotes the instruction for the context-aware translation refinement task, C_{tr} is the document-level context, the same as used in Eq. 3. The result, t_i, is the final, refined translation of a_i.

Once the process of $a_i \Rightarrow t_i$ is finished, all document-level context used in various stages are immediately updated *online* and will be incorporated into the process of $a_{i+i} \Rightarrow t_{i+1}$:

$$\mathcal{C}_{asr} \Rightarrow \mathcal{C}_{asr} = \{s_1, \cdots, s_{i-1}, \underline{s_i}\} \quad (5)$$

$$\mathcal{T}_{tr} \Rightarrow \mathcal{T}_{tr} = \{t_1, \cdots, t_{i-1}, \underline{t_i}\} \quad (6)$$

2.2 Multi-Level Context Integration

The translation of different source sentences requires varying amounts of context (Kang et al., 2020), and the most relevant context for a given segment should be both dynamic and limited in scope. Therefore, using all preceding transcripts and translations as C_{asr} and T_{tr} may be less effective when translating the i-th segment, a_i. To address this limitation, we propose a multi-level context, which

[3] Details of the instructions used in DoCIA can be found in Appendix A.

consists of two components: a short-term context and a long-term context. The multi-level context has a fixed window size $L = m + n$, where m and n represent the number of segments included in the short-term and long-term contexts, respectively.

Short-Memory Context. Related studies (Zhang et al., 2018; Maruf et al., 2019) have shown that adjacent sentences are effective in addressing inter-sentence issues during translation. Hence, we define the short-memory context as the m preceding transcript segments of a_i along with their corresponding translations. Specifically, when translating a_i, the short-memory context consists of the following: the m preceding transcript segments $C^s_{asr} = \{s_{i-m}, \cdots, s_{i-1}\}$ and their corresponding translations $T^s_{tr} = \{t_{i-m}, \cdots, t_{i-1}\}$.

Long-Memory Context. Some clues for alleviating inter-segment issues may lie in a longer memory window (i.e., a window size greater than m), which makes relying solely on the short-memory context insufficient. To address this, we propose incorporating a long-memory context consisting of n transcript segments and the corresponding translations. More specifically, the transcripts and translation in long-memory context are retrieved all preceding segments, except those already included in the short-memory context:

$$C^l_{asr} = f(q_i, \mathcal{C}, n), \tag{7}$$

where $\mathcal{C} = \{s_1, \cdots, s_{i-m}\}$ represents the set of transcripts preceding the short-memory context and f is a retrieval function. Given a query q_i, f returns the top n matching transcript segments from $\mathcal{C}$, forming C^l_{asr}. During ASR refinement, q_i is set to $\bar{s}_i$ while during translation and translation refinement, q_i is set to s_i. Once we obtain C^l_{asr}, we can easily retrieve the corresponding translations T^l_{tr} from $\mathcal{T}_{tr}$. In the strategy, we use BM25 (Lù, 2024) as the retrieval function f. Finally, the document-level context used in Eq. 2,3 and 4 combines long- and short-memory context:

$$C_{asr} = C^l_{asr} + C^s_{asr}, \tag{8}$$

$$T_{tr} = T^l_{tr} + T^s_{tr}. \tag{9}$$

2.3 Refinement Determination Mechanism

To enhance both the overall quality of transcriptions and translations, DoCIA incorporates two context-aware refinement processes. These processes aim to leverage document-level context, improving the coherence and cohesion between segments. Given that inter-segment issues are typically sparse, the refinement process generally focuses on making minor adjustments to the source input. However, excessive refinement could introduce errors that distort the original meaning, leading to hallucinations (Xu et al., 2024b). To address this, we introduce a refinement determination mechanism. Specifically, we define a refinement threshold: if the percentage of modifications in the refined output exceeds this threshold, the refinement is discarded, and the original input is retained as the final output:

$$R = \begin{cases} O & \text{if } g(O, I) \geqslant \lambda, \\ I & \text{if } g(O, I) < \lambda, \end{cases} \tag{10}$$

where I denotes the original input (i.e., the draft text ($\bar{s}_i$ or $\bar{t}_i$), O is the refined output, and R is the final output. λ denotes the threshold of modification. We use the normalized *indel similarity* between I and O as g:

$$g(O, I) = 1 - \frac{d(O, I)}{|I| + |O|}, \tag{11}$$

where $d(\cdot)$ is *Levenshtein edit distance* function, $|\cdot|$ denotes segment length.

For simplicity, we use the same threshold λ for both ASR and translation refinement.

3 Experimentation

In this section, we validate the effectiveness of DoCIA on five ST transation tasks.

3.1 Experimental Settings

Datasets. We conduct our experiments on the MuST-C v1.0 test sets (Di Gangi et al., 2019), which are extracted from TED talks and consist of document-level and sentence/segment-level parallel corpora. In our study, we focus on five language pairs: English (En) $\Rightarrow$ {German (De), Italian(It), Portuguese (Pt), Romanian (Ro), Russian (Ru)}. Each test set contains approximately 2.5K segments drawn from 27 talks (documents).

Metrics. We evaluate translation quality using two COMET-based metrics. For segment-level evaluation, we use s-COMET with the `wmt22-comet-da` model (Rei et al., 2020). For document-level evaluation, we use d-COMET with

System	En ⇒ De		En ⇒ It		En ⇒ Pt		En ⇒ Ru		En ⇒ Ro		*Average*	
	s-Comet	d-Comet	s-Comet	d-Comet	s-Comet	d-Comet	s-Comet	d-Comet	s-Comet	d-Comet	s-Comet	d-Comet
`LLaMA-3.1-8B`												
ASR-SMT	78.01	5.680	79.67	5.619	80.57	5.438	76.36	5.168	79.07	5.372	78.73	5.455
ASR-DMT	77.88	5.712	79.79	5.651	80.69	5.477	76.99	5.211	79.01	5.401	78.87	5.490
DoCIA$_a$	78.11	5.764	80.03	5.703	81.45	5.519	77.16	5.288	79.69	5.473	79.29	5.549
DoCIA$_{a\text{-}m}$	78.50	5.801	80.53	5.792	**81.99**	5.621	78.03	5.401	80.39	5.599	79.89	5.643
DoCIA$_{a\text{-}m\text{-}p}$	**79.15**†	**5.912**	**80.88**†	**5.909**	81.75†	**5.757**	**78.39**†	**5.556**	**80.54**†	**5.734**	**80.15**	**5.774**
`LLaMA-3.1-70B`												
ASR-SMT	81.11	5.997	82.01	5.811	82.03	5.626	80.26	5.686	83.28	5.808	81.73	5.785
ASR-DMT	81.54	6.143	82.36	5.976	82.85	5.745	80.99	5.867	83.15	5.979	82.17	5.942
DoCIA$_a$	81.64	6.098	82.55	5.948	82.53	5.740	81.26	5.803	83.82	5.935	82.36	5.905
DoCIA$_{a\text{-}m}$	**82.69**	6.155	**83.85**	6.132	83.87	5.893	**82.73**	6.034	84.64	6.131	83.57	6.069
DoCIA$_{a\text{-}m\text{-}p}$	82.63†	**6.373**	83.66†	**6.264**	**83.99**†	**6.037**	82.69 †	**6.168**	**85.32**†	**6.365**	**83.66**	**6.241**
`GPT-4o-mini`												
ASR-SMT	82.01	6.001	83.14	5.683	82.56	5.671	82.21	5.827	84.25	5.940	82.83	5.824
ASR-DMT	82.42	6.108	83.52	5.833	83.32	5.943	82.80	5.948	84.82	6.018	83.37	5.970
DoCIA$_a$	82.99	6.174	83.70	6.004	84.03	5.804	82.77	5.935	84.89	6.082	83.68	6.000
DoCIA$_{a\text{-}m}$	**83.75**	6.366	84.54	6.233	**84.57**	6.024	84.10	6.215	85.46	6.213	84.48	6.210
DoCIA$_{a\text{-}m\text{-}p}$	83.64†	**6.444**	**84.76**†	**6.387**	84.51†	**6.297**	**84.32**†	**6.286**	**86.34**†	**6.424**	**84.71**	**6.368**
`GPT-3.5-turbo`												
ASR-SMT	81.51	5.974	81.74	5.732	82.40	5.658	79.21	5.566	82.91	5.644	81.55	5.715
ASR-DMT	81.68	5.977	81.93	5.760	82.53	5.687	79.50	5.611	83.30	5.687	81.78	5.744
DoCIA$_a$	81.70	5.961	82.30	5.705	82.63	5.634	79.57	5.651	83.77	5.601	81.99	5.710
DoCIA$_{a\text{-}m}$	82.93	6.126	83.18	5.838	83.60	5.763	81.71	5.804	84.68	5.891	83.22	5.884
DoCIA$_{a\text{-}m\text{-}p}$	**82.95**†	**6.192**	**83.39**†	**5.997**	**83.90**†	**5.797**	**81.97**†	**5.841**	**85.01**†	**6.033**	**83.45**	**5.973**

Table 1: s-Comet and d-Comet scores on five ST directions when using various LLMs. The column of ***Average*** refers to the averaged performance across all translation directions. The top score in each block is highlighted in **bold** font. Darker colors indicate greater improvements. † indicates that DoCIA$_{a\text{-}m\text{-}p}$ achieves significantly higher s-Comet scores than ASR-SMT/ASR-DMT with a p-value < 0.01.

the `wmt21-comet-qe-mqm` model (Vernikos et al., 2022), which incorporates document-level context to assess improvements across segments.

Models and Hyperparameters. DoCIA is built upon four LLMs: two closed-source models, `GPT-4o-mini` and `GPT-3.5-turbo` (OpenAI, 2023), and two open-source models, `LLaMA-3.1-8B` and `LLaMA-3.1-70B` (Dubey et al., 2024), and run inference of open-source models with 8× Ascend 910B NPUs. For all experiments, we use `Whisper-medium` (Radford et al., 2023) to generate draft ASR results. During inference, we set `do_sample` to true to enable sampling, allowing the LLMs to generate more diverse outputs. A discussion on the impact of different ASR models is provided in Appendix C. We set the context window size L as 6, with $m = n = 3$. The refinement threshold λ is set to 0.7. Further model and hyperparameter selection details are discussed in Appendix C and D.

Comparison System. We implement the following two systems for comparison: 1) **ASR-SMT**, which performs segment-level translation directly on the draft ASR output; 2) **ASR-DMT**, which performs context-aware translation directly on the draft ASR output, using all preceding ASR segments to incorporate document-level context. To better analyze the impact of document-level context at different stages, we define three configurations of DoCIA: 1) **DoCIA**$_a$, which only the context-aware ASR refinement stage; 2) **DoCIA**$_{a\text{-}m}$, which integrates both context-aware ASR refinement and MT; and 3) **DoCIA**$_{a\text{-}m\text{-}p}$, which in all three text-to-text stages, leverages document-level information.

3.2 Main Results

We report our main results in Table 1. Additionally, we report the ASR refinement results in Appendix B. From them, we have the following observations:

DoCIA gains a great improvement over baseline systems. DoCIA delivers substantial gains over both ASR-SMT and ASR-DMT, particularly in d-Comet scores, highlighting its effectiveness in handling document-level context. For example, with the `LLaMA-3.1-8B` model, the configuration DoCIA$_{a\text{-}m\text{-}p}$ (which fully integrates document-level context) achieves an average s-Comet score of 80.15 and a d-Comet score of 5.774. This outperforms both ASR-SMT and ASR-DMT, with improvements of +1.42 in s-Comet and +0.319 in d-Comet over ASR-SMT. Similarly, with the `GPT-4o-mini` model, DoCIA$_{a\text{-}m\text{-}p}$ shows even more pronounced improvements, surpassing ASR-SMT by +1.88 in s-Comet and +0.544 in d-Comet. This demonstrates the effectiveness of incorporating document-level context in ST.

Better base model brings more significant improvement. DoCIA yields more substantial improvements when applied to a better base model such as `LLaMA-3.1-70B` and `GPT-4o-mini`. For instance, with `LLaMA-3.1-8B`, DoCIA results in improvements of +1.42 in s-Comet and +0.319 in d-Comet on average, compared to ASR-SMT. While using `GPT-4o-mini` as the base model, DoCIA achieves even larger gains, with improvements of +1.93 in s-Comet and +0.466 in d-Comet. This may suggest that more powerful LLMs can better utilize document-level context within the DoCIA framework, resulting in improved speech translation quality and enhanced context.

Document-level context boosts performance more when combined with other stages than using alone. When the document-level context is integrated into the ASR refinement phase alone (i.e., DoCIA$_a$), the improvements in s-Comet and d-Comet scores are relatively small but still noticeable. For example, with `LLaMA-3.1-8B`, DoCIA$_a$ shows a modest improvement of +0.56 in s-Comet and +0.094 in d-Comet on average compared to ASR-SMT. However, the performance boost becomes much more substantial when combined with additional stages. For example, compared to DoCIA$_a$ which solely incorporates document-level context during ASR refinement, DoCIA$_{a\text{-}m}$ bring a + 1.12 s-Comet and + 0.164 d-Comet gains. This demonstrates that the multi-stage integration approach effectively unlocks the potential of document-level context, enabling comprehensive optimization of ST.

3.3 Ablation Study

In this section, we conduct an ablation study to evaluate the contributions and impacts of individual components within DoCIA (i.e., DoCIA$_{a\text{-}m\text{-}p}$), including the multi-level context integration and the refinement determination. As shown in Table 2, the comparison shows that the refinement determination (*w/o* R.D.) primarily affects s-Comet, while the multi-level context integration influences d-Comet more. For instance, removing the refinement determination module leads to a 0.98 drop in s-Comet and 0.145 in d-Comet for En⇒De translation using the `GPT-4o-mini` model. While disabling the long-memory context in multi-level context integration (*w/o* L.C.) causes a decrease of 0.63 in s-Comet and 0.243 in d-Comet. This suggests that the two components are complementary, highlighting the necessity of their combined use. Furthermore, we observe that long-memory context has a more substantial effect on performance than short-term context, underscoring the importance of leveraging long-range dependency.

System	En ⇒ De		En ⇒ Ru	
	s-Comet	d-Comet	s-Comet	d-Comet
`LLaMA-3.1-8B`				
DoCIA	79.15	5.912	78.39	5.556
w/o R.D.	78.33	5.812	77.50	5.431
w/o S.C.	78.63	5.792	77.81	5.331
w/o L.C.	78.41	5.761	77.88	5.311
`LLaMA-3.1-70B`				
DoCIA	82.62	6.373	82.69	6.168
w/o R.D.	81.97	6.299	81.81	6.037
w/o S.C.	82.23	6.198	82.11	5.901
w/o L.C.	82.35	6.211	82.19	5.863
`GPT-4o-mini`				
DoCIA	83.64	6.444	84.32	6.286
w/o R.D.	82.66	6.299	83.11	6.116
w/o S.C.	83.11	6.231	83.88	6.061
w/o L.C.	83.01	6.201	83.77	6.011
`GPT-3.5-turbo`				
DoCIA	82.95	6.192	81.97	5.841
w/o R.D.	82.19	6.104	81.01	5.806
w/o S.C.	82.46	6.037	81.23	5.711
w/o L.C.	82.51	6.072	81.35	5.694

Table 2: Ablation study for *refinement determination* (R.D.) and *multi-level context integration*. *w/o S.C.* disables short-memory context, using only the top L matching segments from the long-memory context. *w/o L.C.* disables long-memory context and uses the L preceding segments from short-memory context instead.

4 Discussion and Analysis

In this section, we use the En ⇒ De and En ⇒ Ru tasks, with `LLaMA-3.1-8B` and `GPT-4o-mini`, as representative examples to explore how DoCIA (i.e., DoCIA$_{a\text{-}m\text{-}p}$ in Table 1) enhances ST performance.

4.1 Multi-Dimension Evaluation via GPT-4o

In this section, we extend the evaluation by using `GPT-4o` to assess various discourse phenomena. Specifically, we follow Sun et al. (2024) and ask `GPT-4o` to evaluate the inter-sentence fluency, lexical cohesion errors (LE), and grammatical cohesion errors (GE) in the given translations, using reference translations for comparison. As shown

System	En ⇒ De			En ⇒ Ru		
	Fluency	LE↓	GE↓	Fluency	LE↓	GE↓
LLaMA-3.1-8B						
ASR-SMT	3.01	5.21	4.28	2.89	6.11	4.75
ASR-DMT	3.11	4.32	3.63	3.12	5.28	4.63
DoCIA	**3.76**	**1.98**	**1.42**	**3.71**	**3.32**	**2.29**
GPT-4o-mini						
ASR-SMT	4.35	3.21	2.28	4.01	3.78	2.75
ASR-DMT	4.47	2.01	1.77	4.24	2.61	1.63
DoCIA	**5.16**	**1.01**	**0.79**	**4.98**	**1.33**	**0.82**

Table 3: Evaluation results on test set by GPT-4o.

System	En ⇒ De		En ⇒ Ru	
	s-Comet	d-Comet	s-Comet	d-Comet
LLaMA-3.1-8B				
DoCIA	79.15	5.912	78.39	5.556
w/ offline	78.24	5.783	77.30	5.342
GPT-4o-mini				
DoCIA	83.64	6.444	84.32	6.286
w/ offline	82.81	6.252	83.01	6.095

Table 4: Performance comparison between *online* and *offline* DoCIA on test set.

in Table 3, ASR-DMT outperforms ASR-SMT, demonstrating that integrating inter-segment context significantly reduces lexical and grammatical cohesion errors while improving overall fluency. Notably, DoCIA achieves the best performance on all translation tasks across all three metrics, further highlighting its effectiveness in leveraging inter-segment context.

4.2 Effect of *Online/Offline* Setting

In DoCIA, the document context is updated in real-time during the translation process, following an *online* setting. This means the system continuously updates the context based on the latest translation or ASR outputs, leading to more accurate and coherent translations. In contrast, we also compare this with an *offline* setting, denoted as *offline* DoCIA, which does not update the context during translation. In this case, the system uses only the initial segment-level translation or ASR results, without any real-time updates to the context. Specifically, this corresponds to replacing Eq. 5 and Eq. 6 with initial context: $\mathcal{C}_{asr} = \{\bar{s}_1, \cdots, \bar{s}_i\}$ and $\mathcal{T}_{tr} = \{\bar{t}_1, \cdots, \bar{t}_i\}$, respectively. As shown in Table 4, the *offline* DoCIA shows a significant drop in performance compared to online DoCIA. For example, in the En⇒Ru task using the LLaMA-3.1-8B model, *offline* DoCIA results in a -1.09 decrease in s-Comet score and a -0.214 decrease in d-Comet score. This suggests that DoCIA's performance is highly sensitive to the quality of the context, with real-time updates leading to more accurate and effective context, which in turn significantly improves speech translation quality.

System	En ⇒ De			En ⇒ Ru		
	DA	CE↓	CTE↓	DA	CE↓	CTE↓
LLaMA-3.1-8B						
ASR-SMT	89.7	13.0	16.0	76.7	16.3	20.3
ASR-DMT	90.1	9.5	14.0	78.0	11.5	17.3
DoCIA	**92.3**	**5.5**	**7.5**	**80.7**	**8.5**	**11.5**
GPT-4o-mini						
ASR-SMT	92.5	8.0	12.0	81.3	12.3	15.0
ASR-DMT	92.8	7.3	13.0	82.6	11.1	12.3
DoCIA	**94.7**	**3.3**	**6.0**	**85.0**	**7.3**	**9.5**

Table 5: Results of human evaluation on the test set.

4.3 Human Evaluation

We use the Direct Assessment (DA) (Graham et al., 2017) to evaluate the translation quality of DoCIA and its counterparts. Here, human evaluators compare machine translations with human-produced references in the same language and assign a score from 1 to 100. For each translation direction, we randomly select 4 talks, totaling 312 audio segments, and have two professional translators score the translations from DoCIA, ASR-SMT, and ASR-DMT. Additionally, we report the average counts (per talk) of coherence errors (CE) and content translation errors (CTE) annotated by evaluators. The results, presented in Table 5, show that DoCIA outperforms the others with higher DA scores and fewer CE and CTE scores, providing strong evidence of its effectiveness. For more details of human evaluation, refer to Appendix E.

4.4 Effect of Context Window

In this section, we examine the impact of the context window from two perspectives: 1) varying the context window size L, and 2) exploring different combinations of m and n while keeping L fixed. As shown in Figure 3, increasing the context window size L generally improves performance across all metrics. However, the gains start to diminish when L exceeds 6. Figure 4 illustrates the effects of different m and n combinations. Similar to the trends observed in Section 3.3, we find that reducing the short-memory context (i.e., smaller m) has a more significant impact on s-Comet, while decreasing the long-memory context (i.e., smaller n) affects the d-Comet score more. This further re-

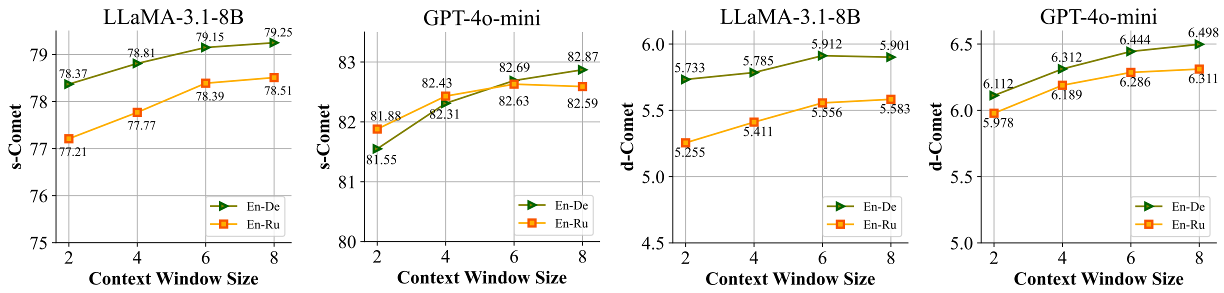


Figure 3: Performance comparison when setting different context window size L.

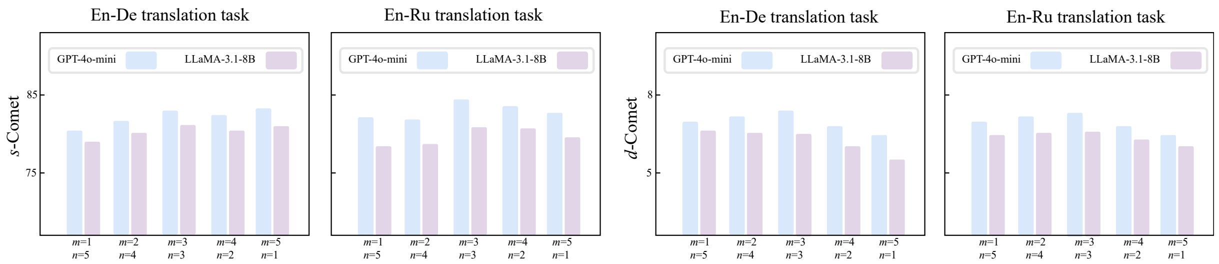


Figure 4: Performance comparison when setting different combinations of m and n.

inforces the complementary nature of short- and long-memory contexts in DoCIA.

5 Related Work

LLM-based Autonomous Agents. LLM-based autonomous agents have recently demonstrated impressive capabilities across a variety of natural language processing tasks. For long-context comprehension and processing, researchers such as Park et al. (2023), Wang et al. (2023a), and Lee et al. (2024) have developed specialized memory and retrieval mechanisms. In efforts to improve output quality, Xu et al. (2024a), Wang et al. (2024), and Feng et al. (2025) have employed prompting techniques that allow LLMs to self-assess and refine their results. Additionally, Li et al. (2023), Liang et al. (2023), Li et al. (2024a), Wu et al. (2024b) and Wang et al. (2025) boost LLM performance on specific tasks through multi-agent collaboration.

Speech-to-Text Translation. Existing studies on ST can be roughly categorized into two groups: cascade-based and end-to-end approaches. The cascade-based system (Zhang et al., 2019a; Sperber and Paulik, 2020; Lam et al., 2021) separates ASR and text translation stages, which doesn't require parallel audio-translation data and can fully leverage ASR and text translation corpus for ST. While the end-to-end system combines these stages and is trained on parallel audio-translation data using strategies such as multi-task learning (Ye et al., 2021), contrastive learning (Ye et al., 2022; Zhang et al., 2022a; Ouyang et al., 2023), sequence mixup (Fang et al., 2022; Yin et al., 2023; Zhang et al., 2023; Zhou et al., 2023), knowledge distillation (Tang et al., 2021; Lei et al., 2023), regularization (Han et al., 2023; Gao et al., 2024), pre-training (Wang et al., 2020; Alinejad and Sarkar, 2020; Tang et al., 2022; Zhang et al., 2022b), and data augmentation (Pino et al., 2019, 2020; Lam et al., 2022). Recently, with the rise of LLMs, some research has explored combining speech encoders with LLMs for end-to-end ST (Wu et al., 2023; Chen et al., 2024). However, few studies explore the effect of document-level information in ST. For example, Tian et al. (2025) enhance ST by incorporating audio context from the two preceding sentences. Similarly, Dou et al. (2025) leverage document-level context during the refinement stage of ST.

Document-Level Text Translation. Document-level context has already been widely considered in text translation studies whether based on the lightweight neural machine translation models (Jean et al., 2017; Wang et al., 2017; Voita et al., 2018; Maruf et al., 2019; Kang et al., 2020; Bao et al., 2021; Sun et al., 2022; Bao et al., 2023) or powerful LLMs (Wang et al., 2023b; Wu and Hu, 2023; Wu et al., 2024a; Li et al., 2024b; Koneru et al., 2024; Lyu et al., 2024). These studies primarily focus on efficiently leveraging document-level context to address inter-sentence translation issues. For example, Lyu et al. (2024) enable LLMs to discriminatively model and utilize both inter- and intra-sentence context, making them more effective at context-aware translation. Similarly, Wu et al. (2024a) investigate effective tuning methods that allow LLMs to better leverage the benefits of document-level context. Despite the effectiveness of document-level context in text translation, it remains underexplored in ST.

6 Conclusion

Inspired by the success of incorporating document-level context in text-to-text MT, we propose DoCIA, an online LLM-based agent designed to improve ST performance by integrating document-level context. DoCIA breaks the whole ST process into four stages, producing the final translation in a cascading manner. Additionally, we introduce a multi-level context integration strategy and a refinement determination mechanism to enhance DoCIA's ability to utilize inter-segment context while minimizing hallucinations during refinement. Experimental results across five ST tasks, using four different LLMs, demonstrate that DoCIA effectively addresses discourse issues from both the ASR and MT stages, leading to significant improvements in overall ST quality.

Acknowledgments

The authors would like to thank the anonymous reviewers for their constructive feedback. This work was supported by the National Natural Science Foundation of China (Grant No. 62036004, 62261160648).

Limitations

In this paper, we propose a document-level context incorporation agent for ST, focusing primarily on its effectiveness in improving ST performance rather than optimizing inference speed. The inference requires multiple calls to LLMs during translation, which results in longer inference latency. Additionally, due to computational resource constraints, DoCIA currently only considers context from the text modality and does not include audio modality information. In the future, we plan to incorporate context from the audio modality to further enhance ST performance.

References

Ashkan Alinejad and Anoop Sarkar. 2020. Effectively pretraining a speech translation decoder with machine translation data. In *Proceedings of EMNLP*, pages 8014–8020.

Guangsheng Bao, Zhiyang Teng, and Yue Zhang. 2023. Target-side augmentation for document-level machine translation. In *Proceedings of ACL*, pages 10725–10742.

Guangsheng Bao, Yue Zhang, Zhiyang Teng, Boxing Chen, and Weihua Luo. 2021. G-transformer for document-level machine translation. In *Proceedings of ACL*, pages 3442–3455.

Zhehuai Chen, He Huang, Andrei Andrusenko, Oleksii Hrinchuk, Krishna C. Puvvada, Jason Li, Subhankar Ghosh, Jagadeesh Balam, and Boris Ginsburg. 2024. Salm: Speech-augmented language model with in-context learning for speech recognition and translation. In *Proceedings of ICASSP*, pages 13521–13525.

Mattia A. Di Gangi, Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2019. MuST-C: a Multilingual Speech Translation Corpus. In *Proceedings of NAACL*, pages 2012–2017.

Huaixia Dou, Xinyu Tian, Xinglin Lyu, Jie Zhu, Junhui Li, and Lifan Guo. 2025. Speech translation refinement using large language models. *Computing Research Repository*, arXiv:2501.15090.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, et al. 2024. The llama 3 herd of models. *Computing Research Repository*, arXiv:2407.21783.

Qingkai Fang, Rong Ye, Lei Li, Yang Feng, and Mingxuan Wang. 2022. STEMM: Self-learning with speech-text manifold mixup for speech translation. In *Proceedings of ACL*, pages 7050–7062.

Zhaopeng Feng, Yan Zhang, Hao Li, Bei Wu, Jiayu Liao, Wenqiang Liu, Jun Lang, Yang Feng, Jian Wu, and Zuozhu Liu. 2025. TEaR: Improving llm-based machine translation with systematic self-refinement. In *Findings of NAACL*, pages 3922–3938.

Pengzhi Gao, Ruiqing Zhang, Zhongjun He, Hua Wu, and Haifeng Wang. 2024. An empirical study of consistency regularization for end-to-end speech-to-text translation. In *Proceedings of ACL*, pages 242–256.

Google. 2023. Palm 2 technical report. *Computing Research Repository*, arXiv:2305.10403.

Yvette Graham, Qingsong Ma, Timothy Baldwin, Qun Liu, Carla Parra, and Carolina Scarton. 2017. Improving evaluation of document-level machine translation quality estimation. In *Proceedings of EACL*, pages 356–361.

Yuchen Han, Chen Xu, Tong Xiao, and Jingbo Zhu. 2023. Modality adaption or regularization? a case study on end-to-end speech translation. In *Proceedings of ACL*, pages 1340–1348.

Sebastien Jean, Stanislas Lauly, Orhan Firat, and Kyunghyun Cho. 2017. Does neural machine translation benefit from larger context? *Computing Research Repository*, arXiv:1704.05135.

Xiaomian Kang, Yang Zhao, Jiajun Zhang, and Chengqing Zong. 2020. Dynamic context selection for document-level neural machine translation via reinforcement learning. In *Proceedings of EMNLP*, pages 2242–2254.

Sai Koneru, Miriam Exel, Matthias Huck, and Jan Niehues. 2024. Contextual refinement of translations: Large language models for sentence and document-level post-editing. In *Proceedings of NAACL-HLT*, pages 2711–2725.

Tsz Kin Lam, Shigehiko Schamoni, and Stefan Riezler. 2021. Cascaded models with cyclic feedback for direct speech translation. In *Proceedings of ICASSP*, pages 7508–7512.

Tsz Kin Lam, Shigehiko Schamoni, and Stefan Riezler. 2022. Sample, translate, recombine: Leveraging audio alignments for data augmentation in end-to-end speech translation. In *Proceedings of ACL*, pages 245–254.

Kuang-Huei Lee, Xinyun Chen, Hiroki Furuta, John Canny, and Ian Fischer. 2024. A human-inspired reading agent with gist memory of very long contexts. In *Proceedings of ICML*, pages 26396–26415.

Yikun Lei, Zhengshan Xue, Xiaohu Zhao, Haoran Sun, Shaolin Zhu, Xiaodong Lin, and Deyi Xiong. 2023. CKDST: Comprehensively and effectively distill knowledge from machine translation to end-to-end speech translation. In *Findings of ACL*, pages 3123–3137.

Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023. CAMEL: communicative agents for "mind" exploration of large language model society. In *Proceedings of NIPS*, pages 51991–52008.

Junkai Li, Siyu Wang, Meng Zhang, Weitao Li, Yunghwei Lai, Xinhui Kang, Weizhi Ma, and Yang Liu. 2024a. Agent hospital: A simulacrum of hospital with evolvable medical agents. *Computing Research Repository*, arXiv:2405.02957.

Yachao Li, Junhui Li, Jing Jiang, and Min Zhang. 2024b. Enhancing document-level translation of large language model via translation mixed-instructions. *Computing Research Repository*, arXiv:2401.08088.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Zhaopeng Tu, and Shuming Shi. 2023. Encouraging divergent thinking in large language models through multi-agent debate. *Computing Research Repository*, arXiv:2305.19118.

Xinglin Lyu, Junhui Li, Yanqing Zhao, Min Zhang, Daimeng Wei, Shimin Tao, Hao Yang, and Min Zhang. 2024. DeMPT: Decoding-enhanced multi-phase prompt tuning for making LLMs be better context-aware translators. In *Proceedings of EMNLP*, pages 20280–20295.

Xing Han Lù. 2024. Bm25s: Orders of magnitude faster lexical search via eager sparse scoring. *Computing Research Repository*, arXiv:2407.03618.

Sameen Maruf, André F. T. Martins, and Gholamreza Haffari. 2019. Selective attention for context-aware neural machine translation. In *Proceedings of NAACL*, pages 3092–3102.

OpenAI. 2023. Gpt-4 technical report. *Computing Research Repository*, arXiv:2303.08774.

Siqi Ouyang, Rong Ye, and Lei Li. 2023. WACO: Word-aligned contrastive learning for speech translation. In *Proceedings of ACL*, pages 3891–3907.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pages 1–22.

Juan Pino, Liezl Puzon, Jiatao Gu, Xutai Ma, Arya D. McCarthy, and Deepak Gopinath. 2019. Harnessing indirect training data for end-to-end automatic speech translation: Tricks of the trade. In *Proceedings of IWSLT*.

Juan Pino, Qiantong Xu, Xutai Ma, Mohammad Javad Dousti, and Yun Tang. 2020. Self-Training for End-to-End Speech Translation. In *Proceedings of Interspeech*, pages 1476–1480.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine Mcleavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *Proceedings of ICML*, pages 28492–28518.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of EMNLP*, pages 2685–2702.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of ACL*, pages 1715–1725.

Matthias Sperber and Matthias Paulik. 2020. Speech translation and the end-to-end promise: Taking stock of where we are. In *Proceedings of ACL*, pages 7409–7421.

Yirong Sun, Dawei Zhu, Yanjun Chen, Erjia Xiao, Xinghao Chen, and Xiaoyu Shen. 2024. Instruction-tuned llms succeed in document-level mt without fine-tuning–but bleu turns a blind eye. *Computing Research Repository*, arXiv:2410.20941.

Zewei Sun, Mingxuan Wang, Hao Zhou, Chengqi Zhao, Shujian Huang, Jiajun Chen, and Lei Li. 2022. Rethinking document-level neural machine translation. In *Findings of ACL*, pages 3537–3548.

Yun Tang, Hongyu Gong, Ning Dong, Changhan Wang, Wei-Ning Hsu, Jiatao Gu, Alexei Baevski, Xian Li, Abdelrahman Mohamed, Michael Auli, and Juan Pino. 2022. Unified speech-text pre-training for speech translation and recognition. In *Proceedings of ACL*, pages 1488–1499.

Yun Tang, Juan Pino, Xian Li, Changhan Wang, and Dmitriy Genzel. 2021. Improving speech translation by understanding and learning from the auxiliary text translation task. In *Proceedings of ACL*, pages 4252–4261.

Xinyu Tian, Haoran Wei, Zhengxian Gong, Junhui Li, and Jun Xie. 2025. Improving end-to-end speech-to-text translation with document-level context. *IEEE Transactions on Audio, Speech and Language Processing*, 33:2098–2109.

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 WMT*, pages 118–128.

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.

Bing Wang, Xinnian Liang, Jian Yang, Hui Huang, Shuangzhi Wu, Peihao Wu, Lu Lu, Zejun Ma, and Zhoujun Li. 2023a. Enhancing large language model with self-controlled memory framework. *Computing Research Repository*, arXiv:2304.13343.

Chengyi Wang, Yu Wu, Shujie Liu, Ming Zhou, and Zhenglu Yang. 2020. Curriculum pre-training for end-to-end speech translation. In *Proceedings of ACL*, pages 3728–3738.

Longyue Wang, Chenyang Lyu, Tianbo Ji, Zhirui Zhang, Dian Yu, Shuming Shi, and Zhaopeng Tu. 2023b. Document-level machine translation with large language models. In *Proceedings of EMNLP*, pages 16646–16661.

Longyue Wang, Zhaopeng Tu, Andy Way, and Qun Liu. 2017. Exploiting cross-sentence context for neural machine translation. In *Proceedings of EMNLP*, pages 2826–2831.

Yutong Wang, Jiali Zeng, Xuebo Liu, Fandong Meng, Jie Zhou, and Min Zhang. 2024. TasTe: Teaching large language models to translate through self-reflection. In *Proceedings of ACL*, pages 6144–6158.

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 *Proceeding of ICLR*.

Jian Wu, Yashesh Gaur, Zhuo Chen, Long Zhou, Yimeng Zhu, Tianrui Wang, Jinyu Li, Shujie Liu, Bo Ren, Linquan Liu, and Yu Wu. 2023. On decoder-only architecture for speech-to-text and large language model integration. In *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1–8.

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

Minghao Wu, Yulin Yuan, Gholamreza Haffari, and Longyue Wang. 2024b. (perhaps) beyond human translation: Harnessing multi-agent collaboration for translating ultra-long literary texts. *Computing Research Repository*, arXiv:2405.11804.

Yangjian Wu and Gang Hu. 2023. Exploring prompt engineering with GPT language models for document-level machine translation: Insights and findings. In *Proceedings of WMT*, pages 166–169.

Wenda Xu, Daniel Deutsch, Mara Finkelstein, Juraj Juraska, Biao Zhang, Zhongtao Liu, William Yang Wang, Lei Li, and Markus Freitag. 2024a. LLMRefine: Pinpointing and refining large language models via fine-grained actionable feedback. In *Findings of NAACL*, pages 1429–1445.

Ziwei Xu, Sanjay Jain, and Mohan S. Kankanhalli. 2024b. Hallucination is inevitable: An innate limitation of large language models. *Computing Research Repository*, arXiv:2401.11817.

Rong Ye, Mingxuan Wang, and Lei Li. 2021. End-to-end speech translation via cross-modal progressive training. In *Proceedings of INTERSPEECH*, pages 2267–2271.

Rong Ye, Mingxuan Wang, and Lei Li. 2022. Cross-modal contrastive learning for speech translation. In *Proceedings of NAACL*, pages 5099–5113.

Wenbiao Yin, Zhicheng Liu, Chengqi Zhao, Tao Wang, Jian Tong, and Rong Ye. 2023. Improving speech translation by fusing speech and text. In *Findings of EMNLP*, pages 6262–6273.

Hao Zhang, Nianwen Si, Yaqi Chen, Zhen Li, Tong Niu, Xukui Yang, and Dan Qu. 2022a. FCGCL: Fine- and coarse-granularity contrastive learning for speech translation. In *Findings of EMNLP*, pages 3048–3059.

Jiacheng Zhang, Huanbo Luan, Maosong Sun, Feifei Zhai, Jingfang Xu, Min Zhang, and Yang Liu. 2018. Improving the transformer translation model with document-level context. In *Proceedings of EMNLP*, pages 533–542.

Linlin Zhang, Kai Fan, Boxing Chen, and Luo Si. 2023. A simple concatenation can effectively improve speech translation. In *Proceedings of ACL*, pages 1793–1802.

Pei Zhang, Niyu Ge, Boxing Chen, and Kai Fan. 2019a. Lattice transformer for speech translation. In *Proceedings of ACL*, pages 6475–6484.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2019b. Bertscore: Evaluating text generation with bert. In *Proceedings of ICLR*.

Ziqiang Zhang, Long Zhou, Junyi Ao, Shujie Liu, Lirong Dai, Jinyu Li, and Furu Wei. 2022b. SpeechUT: Bridging speech and text with hidden-unit for encoder-decoder based speech-text pre-training. In *Proceedings of EMNLP*, pages 1663–1676.

Yan Zhou, Qingkai Fang, and Yang Feng. 2023. CMOT: Cross-modal mixup via optimal transport for speech translation. In *Proceedings of ACL*, pages 7873–7887.

A Prompt Templates in DoCIA

This section presents the prompt templates used in each stage of DoCIA. The prompt templates for ASR Refinement, machine translation and translation refinement are shown in Figure 5, 6 and 7, respectively. To ensure accuracy and proper formatting, we instruct the LLM to generate outputs in JSON format.

B Performance of ASR Refinement

Refining Model	WER↓	ERA	BERTScore
LLaMA-3.1-8B	19.16	55.81	88.75
LLaMA-3.1-70B	18.66	56.45	88.97
GPT-4o-mini	16.01	**57.12**	**89.21**
GPT-3.5-turbo	18.96	56.87	89.05
Draft ASR	**14.71**	55.78	88.50

Table 6: Performance comparison of ASR refinement when using various LLMs.

In this section, we evaluate the performance of ASR refinement in addition to the main translation performance. Apart from Word Error Rate (WER),[4] we compare BERTScore (Zhang et al., 2019b) and Entity Recognition Accuracy (ERA) to assess how well the models utilize context to improve semantic accuracy and correct entity recognition errors. ERA is evaluated using GPT-4o. Specifically, we first use GPT-4o to extract entities from both the refined and non-refined ASR outputs (draft ASR), as well as from the reference ASR. ERA is calculated by comparing the extracted entities to the reference.

As shown in Table 6, although WER increases after refinement, both ERA and BERTScore show improvements. This indicates that leveraging document-level context significantly enhances entity recognition and semantic accuracy.

Additionally, we present a case study in Figure 8, where DoCIA corrects an ASR error. In this case, "*DigiNotar*" is misrecognized as "*TAR*" in draft ASR, but DoCIA successfully corrects the error by considering the inter-segment context, which include the proper entity "*DigiNotar*".

[4] In this paper, we retain the punctuation from the ASR results and report the case-sensitive WER.

C Effect of ASR Model

System	En ⇒ De		En ⇒ Ru	
	s-Comet	d-Comet	s-Comet	d-Comet
LLaMA-3.1-8B				
ASR-SMT	78.01	5.680	76.36	5.168
ASR-DMT	77.88	5.712	76.99	5.211
DoCIA (*w/ WM*)	**79.15**	5.912	78.39	5.556
w/ WS, WER=14.89	78.99	5.901	78.45	**5.562**
w/ WL, WER=14.41	79.11	**5.935**	**78.61**	5.551
GPT-4o-mini				
ASR-SMT	82.01	6.001	82.21	5.827
ASR-DMT	82.42	6.108	82.80	5.948
DoCIA (*w/ WM*)	83.64	6.444	84.32	6.286
w/ WS, WER=14.89	83.43	6.401	84.34	6.275
w/ WL, WER=14.41	**83.82**	**6.478**	**84.71**	**6.299**

Table 7: Performance comparison when using various ASR models. **WS**, **WM** and **WL** denote the Whisper-Small, Whisper-Medium and Whisper-Large models, respectively.

In our experiments, DoCIA uses the Whisper-Medium ASR model to generate segment-level transcriptions. We now investigate the effect of using ASR models of different sizes on the final translation performance. Table 7 presents a comparison of translation results across different ASR models. It shows that larger ASR models tend to achieve better ASR performance (i.e., lower WER), leading to modest improvements in translation quality. For instance, using the Whisper-Large yields a +0.39 improvement in the s-COMET score for the En⇒Ru task compared to the Whisper-Medium, when DoCIA uses the GPT-4o-mini translation model.

D Effect of Hyperparameter λ in Refinement Determination

To prevent hallucinations in both the transcription and translation refinement processes, we introduce a refinement determination mechanism. In this

Context-aware ASR Refinement Prompt Template

You are an expert in automatic speech recognition refinement. Given an automatic speech recognition sentence in `<SRC-LANG>`, please check it based on its preceding automatic speech recognition sentences. Correct the capitalization, add punctuation, and eliminate incoherences such as fillers, false starts, repetitions, corrections, hesitations, and interjections. Maintain the original meaning and structure of the sentence and make it more coherent with the preceding ASR sentence. Provide your output in the following JSON format:
{'Output': `<Refined ASR sentence>`}

Preceding ASR sentences:
`<Preceding ASR sentence>`

Draft current ASR sentence:
`<Draft current ASR sentence>`

Your output:

Figure 5: Prompt template for ASR Refinement in DoCIA.

Context-aware Translation Prompt Template

You are a professional translator from `<SRC-LANG>` to `<TGT-LANG>`. Given a current source sentence, please translate it to `<TGT-LANG>` based on its preceding source sentence and translation history. The translation of the current sentence should be more coherent with its preceding translations and have better lexical cohesion. Provide your translation in the following JSON format:"
{'Output': `<Translation>`}

Preceding source sentences:
`<Preceding source sentences>`

Preceding translation history:
`<Preceding translation history>`

Current source sentence:
`<Current source sentence>`

Your output:

Figure 6: Prompt template for context-aware translation in DoCIA.

section, we investigate the impact of the determination threshold, λ, and explore the effect of using BERTScore to compute the similarity between I (input) and O (output) by replacing Eq. 11 with BERTScore. The results, presented in Table 8, show that both excessively high and low threshold values negatively affect performance. Additionally, using BERTScore in the refinement determination process leads to significant performance improvements. This suggests that the determination mechanism is not highly sensitive to the choice of similarity function.

E Details of Human Evaluation

Recruitment and Criterion. We recruit evaluators who are professional translators with a mini-

Context-aware Translation Refinement Prompt Template

You are a professional <SRC-LANG> to <TGT-LANG> translation post-editor. Given a current source sentence and its draft translation, please refine the draft translation based on its preceding source sentence and translation history. The refined translation should have the same semantics as the current source sentence be more coherent and have better lexical cohesion with its preceding translation history. Provide the refined translation in the following JSON format:
{'Output': `<Refined Translation>`}

Preceding source sentences:
`<Preceding source sentences>`

Preceding translation history:
`<Preceding translation history>`

Current source sentence:
`<Current source sentence>`

Draft translation:
`<Draft translation>`

Your output:

Figure 7: Prompt template for context-aware translation refinement in DoCIA.

Inter-Segment Context

And then we look at cases like what happened in DigiNotar. This is a prime example of what happens when governments attack against their own citizens.

ASR Result of Current Segment

Did you know **TAR** is a certificate authority from the Netherlands? Or actually it was?

Refined ASR Result by DoCIA

DigiNotar is a certificate authority from the Netherlands. Or actually, it was.

Reference ASR Result

DigiNotar is a certificate authority from the Netherlands —or actually, it was.

Figure 8: A case study for context-aware ASR refinement. ASR result is from `Whisper-Medium`.

System	En ⇒ De		En ⇒ Ru	
	s-Comet	*d*-Comet	*s*-Comet	*d*-Comet
`LLaMA-3.1-8B`				
DoCIA ($\lambda = 0.7$)	**79.15**	**5.912**	78.39	**5.556**
$\lambda = 0.0$	78.24	5.735	77.56	5.441
$\lambda = 0.5$	78.54	5.733	77.81	5.533
$\lambda = 0.9$	78.21	5.712	77.31	5.432
$\lambda = 1.0$	78.33	5.812	77.50	5.431
w/ BS ($\lambda = 0.7$)	78.81	5.865	**78.45**	5.511
`GPT-4o-mini`				
DoCIA ($\lambda = 0.7$)	83.64	**6.444**	**84.32**	**6.286**
$\lambda = 0.0$	82.79	6.259	83.33	6.199
$\lambda = 0.5$	83.31	6.387	83.83	6.218
$\lambda = 0.9$	82.98	6.253	83.45	6.166
$\lambda = 1.0$	82.66	6.299	83.11	6.116
w/ BS ($\lambda = 0.7$)	**83.75**	6.393	84.12	6.201

Table 8: Performance comparison when setting different λ. When setting $\lambda = 1.0$ (or $\lambda = 0.0$), we always take the original (or refined) text as the final output.

mum of five years of experience. Given a reference ASR output, its translations from various systems, and the human-produced reference translation, evaluators are tasked with assigning a score on a scale from 0 to 100. The detailed scoring criterion as follows:

- `0-20`: The translation is completely incorrect and unclear, with only a few words or phrases being correct. It is totally unreadable and difficult to understand.
- `21-40`: The translation has very little semantic similarity to the source sentence, with key information missing or incorrect. It has numerous unnatural and unfluent expressions and grammatical errors.
- `41-60`: The translation can express part of the

key semantics but has many non-key semantic errors. It lacks fluency and idiomaticity.

- `61-80`: The translation can express the key semantics but has some non-key information errors and significant grammatical errors. It lacks idiomaticity.
- `81-100`: The translation can express the semantics of the source sentence with only a few non-key information errors and minor grammatical errors. It is fluent and idiomatic.

Coherence Error and Content Error. We manually count the average number of coherence errors (CE) and content translation errors (CTE) for evaluation terms. Specifically, CE involves two types of errors, including inter-sentential consistency errors, such as inconsistent translations of the same entity across sentences, and inter-sentential logical errors, such as improper translation or usage of transition words and conjunctions. CTE includes three error types: mistranslation, under- and over-translation.