Overview of the SIGTURK 2026 Shared Task: Terminology-Aware Machine Translation for English–Turkish Scientific Texts

Ali Gebeşce[1,2,*], **Abdulfattah Safa**[1,2,*], **Ege Uğur Amasya**[1], **Gözde Gül Şahin**[1,2,3]
[1]Computer Engineering Department, Koç University, Istanbul, Turkey
[2] KUIS AI Lab, Istanbul, Turkey
[3] FAU Erlangen-Nürnberg, Erlangen, Germany
https://gglab-ku.github.io/

Abstract

This paper presents an overview of the SIGTURK 2026 Shared Task on Terminology-Aware Machine Translation for English-Turkish Scientific Texts. We address the critical challenge of terminological accuracy in low-resource settings by constructing the first terminology-rich English-Turkish parallel corpus, comprising 3,300 sentence pairs from STEM domains with 10,157 expert-validated term pairs. The shared task consists of three subtasks: term detection, expert-guided correction, and end-to-end post-editing. We evaluate state-of-the-art baselines (including GPT-5.2 and Claude Sonnet 4.5) alongside participant systems employing diverse strategies from fine-tuning to Retrieval-Augmented Generation (RAG). Our results highlight that while massive generalist models dominate zero-shot detection, smaller, domain-adapted models using Supervised Fine-Tuning and Reinforcement Learning can significantly outperform them in end-to-end post-editing. Furthermore, we find that rigid retrieval pipelines often disrupt fluency, whereas Chain-of-Thought prompting allows models to integrate terminology more naturally. Despite these advances, a significant gap remains between automated systems and human expert performance in strict terminology correction.

1 Introduction

Automatic translation systems routinely stumble over technical terms, yet those very terms are critical for knowledge transfer in science, engineering, and mathematics. Terminological errors can obscure meaning, slow down human post-editing, and erode trust in machine-generated output. Despite advanced capabilities of large language models, following rigid constraints and instructions, such as following a strict terminology database, remains an unsolved problem for many language pairs.

Researchers organized several shared tasks (Alam et al., 2021; Semenov et al., 2023, 2025) on terminology-aware translation. They mostly focuse on several high to mid resource languages such as English, French, Korean and Czech; and, treated terms and translations *as given*. On the other hand, this shared task focuses on a relatively underexplored, English to Turkish direction, and proposes a more challenging task of end-to-end post-edit with an offline glossary without providing the term boundaries.

To enable such tasks, we first create a high-quality terminology-rich corpus based on the Mathematics, Physics, and Computer Science articles in Turkish Wikipedia and abstracts of theses published at Turkish National Thesis Center in the same fields. Next, we annotate the corpus with technical terms, their links to the terminology dictionary, terimler.org, and correct their translations based on the dictionary (if necessary). 43 trained annotators achieve substantial agreement (Fleiss $\kappa \simeq 0.71$ for English and 0.67 for Turkish term detection) while earning above-market wages, resulting in 3,300 sentence pairs from 590 paragraphs, with 10,157 expert-validated term links and corrections. Finally, we define three subtasks on the annotated corpus, namely as: term detection, term correction and end-to-end post-editing.

The shared task evaluated five state-of-the-art zero-shot baselines: GPT-5.2, Claude Sonnet 4.5, Gemini 3 Flash, DeepSeek v3.2, and Llama 4 Scout across all three subtasks. We received submissions from four teams employing a diverse range of strategies, including the use of Supervised Fine-Tuning (SFT) and Reinforcement Learning (RL) to adapt compact 4B-parameter models. Other approaches included modular Retrieval-Augmented Generation (RAG) pipelines with fuzzy glossary retrieval, multi-step Chain-of-Thought (CoT) reasoning frameworks for precise term positioning, and iterative refinement techniques using paragraph-

[*]Equal contribution.

level context to ensure terminological consistency. We find that for term detection, top-performing models match or even exceed human annotator performance, while term correction proves most difficult. This is due to models tending to revert to familiar terminology despite explicit expert guidance, leaving a significant gap compared to human experts. For end-to-end post-editing, domain-adapted fine-tuning proves most effective, enabling smaller specialized models to outperform the strongest generalist baselines. To facilitate further research, all data[1], code, and guidelines[2] are publicly available.

2 Related Work

The first shared task on terminology translation was organized by WMT 2021 (Alam et al., 2021) and focused on COVID-19 terms for several high to mid resource languages such as English, French, Korean and Czech. WMT 2023 (Semenov et al., 2023) shared task on terminology translation has added "random terms" along with the "proper terms" and evaluated the ability of the systems to distinguish between the two. Finally, WMT 2025 has broadened the task to cover more domains and language pairs, explicitly evaluating how systems exploit external dictionaries (Semenov et al., 2025). Outside WMT, researchers have released robustness suites that stress different constraint lengths and densities (Zhang et al., 2023), and domain-specific evaluations in medical MT have shown that even state-of-the-art transformers still mistranslate up to 18% of critical terms when training data are scarce (Dogru, 2021). Unlike previous shared tasks, we define an end-to-end task where the systems first need to identify the term boundaries and then exploit an external dictionary to translate the detected terms; we focus on three domains (Math, Informatics, and Physics) for the English to Turkish direction.

3 Dataset

We leverage two resources to create the dataset: Wikipedia Content Translation Tool Dump; and Turkish National Thesis Center Dataset. First, we describe the corpus creation procedure §3.1, then we define the annotation process §3.2 followed by the final dataset statistics.

[1] https://github.com/GGLAB-KU/twist

[2] https://github.com/GGLAB-KU/sigturk2026_sharedtask

3.1 Corpus

Wikipedia Content Translation Dump The Wikipedia Content Translation tool simplifies the translation process by automating repetitive tasks such as copying text, creating links, and categorizing articles. Translated paragraph pairs are published weekly in Wikimedia dumps[3]. We start with the June 7, 2024 dump, which contains 468,254 parallel paragraphs in English and Turkish. Then we remove the empty, duplicate, and identical content; and exclude insufficient or inconsistent lengths and eliminate content containing symbols like ↑, &, or *displaystyle*, which often refer to references or embedded equations in Wikipedia pages. Since we focus on technical terms, we restrict the articles to three STEM domains: Mathematics, Physics, and Computer Science. Fields such as Chemistry and Biology are excluded due to limited domain expertise. To have a terminologically rich corpus, we retain only the paragraphs where the number of unique terms exceeds three. Next, we align English-Turkish sentences and filter out those where the number of source sentences does not match the number of target sentences. We then use the GPT-4o model [4] to filter the paragraphs outside the chosen domains. Finally, we manually review the remaining paragraphs and eliminate instances with poor translation which reduces the corpus to 303 paragraphs containing a total of 1,185 sentences.

Turkish National Thesis Dataset The Turkish National Thesis Center [5], managed by the Turkish Council of Higher Education (YÖK), is the official repository for graduate theses from Turkish universities, contains over 700,000 theses. We select abstracts exclusively from theses in the Mathematics, Physics, and Computer Science departments. From six universities [6], we compile 287 abstracts comprising 2,115 sentences. Since the theses are submitted to the Turkish National Thesis Center in PDF format, OCR-related typos occasionally occur in the abstracts. To address this, we use the GPT-4o model to correct these typos.

3.2 Annotation

We combine sentences from both dataset, resulting in a total of 3,300 sentences, evenly distributed

[3] https://dumps.wikimedia.org/other/contenttranslation/

[4] https://openai.com/index/gpt-4o-system-card/

[5] https://tez.yok.gov.tr/UlusalTezMerkezi

[6] Koç University, Middle East Technical University, Istanbul Technical University, Bilkent University, Boğaziçi University, and Sabancı University

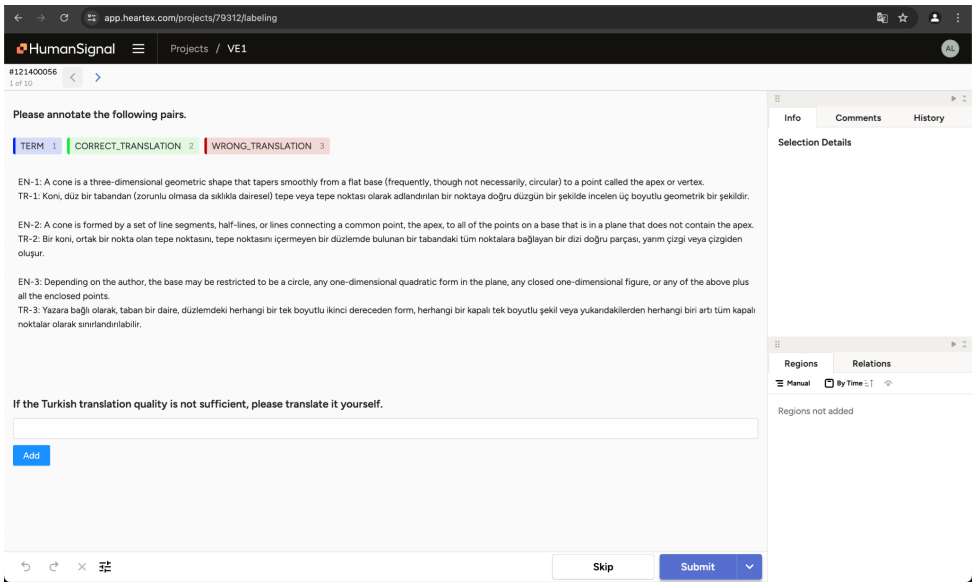

Figure 1: Label Studio Annotation Interface

across the three domains: Mathematics, Physics, and Computer Science, with each domain contributing 1,100 sentences. We use Label Studio's Academic Program [7] due to its free access, online functionality, and user-friendly interface for annotation. The interface is given in Fig. 1.

Our annotation guideline provides step-by-step instructions for the annotation process, including a FAQ on handling special cases (e.g., terms connected with "and", abbreviations, terms containing suffixes) during annotation. The annotation steps summarized in Algorithm 1 begins by identifying English terms in the source sentence, which are labeled as terms. Corresponding Turkish terms are then identified in the target sentence and labeled as terms as well. For each English-Turkish term pair, a relation is established to link the terms. These relations are validated using the `terimler.org` terminology database, where correctly translated terms are marked as `CORRECT_TRANSLATION`, and incorrect ones are flagged with updated metadata. If an English term does not have a match in the database, its Turkish counterpart is manually evaluated and labeled as either correct or incorrect.

Additionally, the Annotation Guideline includes a complementary section titled Special Cases, which provides annotators with instructions for handling unique scenarios. These scenarios include terms connected with "and", abbreviations, terms containing suffixes, and other such cases requiring special treatment. There are 10 special cases outlined in this section, and their summaries appear in Table 3 in App A.

Annotators We use various channels to identify potential annotators, including university faculty mailing lists, WhatsApp, Telegram, and Discord groups associated with universities, LinkedIn, Twitter, and science platforms such as *fizikhaber.com* and *Türk Fizik Takvimi*. Next, we conduct an Annotation Webinar with potential annotators to introduce the project and explain key topics, including the project's overview, the role of annotators, the quiz process and scoring system, annotation guidelines, the quiz application form, the project timeline, etc... After the webinar, we perform a quiz among interested parties to proceed further. The quiz consists of 30 pairs of sentences, organized into 10 instances of three sentences each. Out of the 160 participants who received the quiz, 84 completed all 30 sentences. Next, we eliminate participants with an F1 score below 0.7, leaving 49 participants who proceed to the annotation phase. Finally, we send an automatically generated Quiz Evaluation Report to participants to help them avoid repeating mistakes during labeling and to improve labeling quality. All annotators have at least an undergraduate degree (or are senior undergraduate students), and the majority are in the 18–25 age group. In terms of academic specialization, 18 annotators are from computer science, 8 from mathematics, and 6 from physics.

Annotation Design: After the quiz, we start the annotation process with the remaining high quality

[7] https://labelstud.io/academic/

annotators. First, we manually create a test set of 300 randomly selected sentences (around 10% of the dataset). Annotators see a randomly selected test question at each annotation page for continuous performance evaluation. The order of test instances is randomized. If an annotator's performance falls below a certain threshold an automated email is sent to halt their work. We then review the annotations to determine whether an annotator is inattentive or spamming.

The annotation process is evaluated across four subtasks: English-Turkish Term Detection (F1 score), Translation Labeling (Accuracy), Translation Correction (Exact Match), and Term Linking (Exact Match). Partial credits are given for partial solutions as explained at the shared task repo. The annotation results demonstrate robust performance across tasks, with all metrics improving compared to the quiz. English term detection achieves an F1 score of 0.84 and Cohen's Kappa (Cohen, 1960) of 0.81, while Turkish term detection scores 0.82 and 0.77, respectively. Turkish translation labeling and correction achieve an exact match score of 0.85, and 0.68, while term linking maintains strong performance with a score of 0.86. Finally, we calculate Fleiss Kappa (Fleiss, 1971) to assess inter-annotator agreement for English and Turkish term detection as 0.715, indicating substantial agreement among annotators. Similarly, for Turkish term detection, the mean Fleiss Kappa score is 0.674, reflecting moderate to substantial agreement.

Annotation Cost The payment per sentence is 20 Turkish Liras [8], with the average time to annotate one sentence approximately 4 minutes. Each of the 3,300 unique sentences is annotated by three annotators. A total of 9,900 sentences are annotated, resulting in a total annotation cost of 198,000 TL. Annotators earn an average hourly rate of 300 TL, significantly higher than Turkey's minimum net hourly wage of 75.56 TL. The number of sentences annotated per annotator varies, with a mean of 230.23 sentences. Additionally, bonus payments totaling 15,000 TL are awarded to 18 annotators for providing high-quality comments to identify errors in `terimler.org`.

3.3 Post Annotation

As one of the goals is to provide feedback to the terminology database, we collected 2,100 comments from the annotators. After cleaning, we aggregated 294 unique entries for `terimler.org`, including 214 entries suggesting synonyms, 37 identifying typos, 22 highlighting potential errors in meaning, and 21 pointing out definite errors requiring correction.

[8] Annotation is conducted during June 2025

Algorithm 1: Annotation Steps

Input: English and Turkish sentence pair $(S_{\text{EN}}, S_{\text{TR}})$
Output: Annotated terms $\mathcal{T}$ and relations $\mathcal{R}$
Step 1: Find English Terms: Label each $t_{\text{EN}} \in S_{\text{EN}}$ with `TERM`.
Step 2: Find Turkish Pairs: For each t_{EN}, label corresponding $t_{\text{TR}} \in S_{\text{TR}}$ with `TERM`.
Step 3: Create Relations: For each pair $(t_{\text{EN}}, t_{\text{TR}})$, create a relation (arrow).
Step 4: Validate Translations:
foreach $(t_{EN}, t_{TR}) \in \mathcal{R}$ **do**
  if t_{TR} *is correct on* `terimler.org` **then**
    Label as `CORRECT_TRANSLATION` and add metadata;
  else
    Label as `WRONG_TRANSLATION` and update metadata;
Step 5: Handle Missing Terms:
For t_{EN} not in `terimler.org`, evaluate t_{TR} and label as correct or incorrect.

We define **Gold Terms** as those tagged by all three annotators; and **Silver Terms** as those tagged by two out of three annotators. Final annotated dataset (Gebeşçe et al., 2025) contains 5,845 gold terms and 2,625 silver terms across 3,300 English-Turkish parallel sentences. We manually review the remaining terms without any agreement, and perform expert aggregation. In total, the dataset contains 10,157 aligned terms annotated with terminology links and post editing information.

4 Shared Task

The SIGTURK 2026 Shared Task on Terminology-Aware Machine Translation explores whether models can follow domain experts' translation choices and automatically correct or post-edit translations accordingly. The shared task consists of three subtasks, each addressing a different aspect of terminology-aware translation.

Subtask 1: Term Detection Given parallel English-Turkish sentence pairs along with their sur-

rounding paragraph context, systems must identify the boundaries of technical terms in both languages and align them as term pairs. We perform token-based precision, recall, and micro/macro F1 based on span overlaps.

Subtask 2: Term Correction with Expert Input. This subtask focuses on post-editing the translation of technical terms using expert-provided hints. Given the detected term boundaries and expert hints (which may be base forms without suffixes or partial translations), systems must produce morphologically correct Turkish translations that conform to expert terminology preferences. We use Exact Match as the evaluation score.

Subtask 3: End-to-End Post-Edit. This subtask evaluates end-to-end performance when systems have access to the `terimler.org` terminology database which is provided as an offline glossary. Without explicit term boundaries or hints, systems must post-edit target sentences to align with standard Turkish scientific terminology.We use chrF (Popović, 2015) and BLEU (Papineni et al., 2002).

Subtask	Dev	Test
1: Term Detection	500	2,800
2: Term Correction	250	780
3: End-to-End Post-Edit	250	780

Table 1: Dataset statistics for the shared task.

Participation and Data Participants may submit systems for any combination of the three subtasks. Table 1 outlines the data splits for each subtask. Due to the nature of the data, a moderate portion of the term translations already complies with the terminology database. For subtasks 2 and 3, we retain only the terms that require correction, resulting in a significantly smaller test set. We do not restrict the usage of external datasets. Furthermore, participants are permitted to use the development set to tune their prompts or fine-tune their models. Finally, it should be noted that Subtasks 1 and 2 can be directly compared with the human performance, while Subtask3 cannot. These subtasks, along with the evaluation scripts, are available in CodeBench[9] and in our GitHub repositories.

[9] https://www.codabench.org/competitions/11661/

5 Systems

5.1 Baseline Models

We evaluate five state-of-the-art language models representing diverse architectural approaches: OpenAI **GPT-5.2**[10], Anthropic **Claude Sonnet 4.5**[11], Google **Gemini 3 Flash**[12], DeepSeek **v3.2**[13], and Meta **Llama 4 Scout**[14].

All models operate with a temperature of 1.0 and a maximum token limit of 8,192. We optimized our system prompts using a small subset of the development data to refine instruction wording and ensure strict adherence to the required output formats; no task-specific examples or labeled instances are included in the final prompts. Complete prompt templates are provided in App. B.

Subtask 1 (Term Detection). The model receives the source sentence, the full source paragraph for context, and the target translation. The prompt guides the model to identify technical terms in the English source and align them with their corresponding Turkish spans.

Subtask 2 (Term Correction). The model receives the list of identified term pairs along with expert hints. It is instructed to generate morphologically correct Turkish replacements that conform to the expert terminology.

Subtask 3 (End-to-End Post-Edit). We use a direct zero-shot approach where the model employs the sentence pair and paragraph context to refine the full Turkish translation. The model is tasked with aligning the output with standard scientific terminology without access to explicit term boundaries.

5.2 Participant Systems

We received submissions from four teams, each employing distinct strategies ranging from fine-tuning to retrieval-augmented pipelines. None of the teams submitted a system paper, however, they submitted their system descriptions and results to our evaluation platform.

New Mind AI Research. This team focused on specific model adaptation by fine-tuning Qwen3-4B-Instruct[15] for subtask 3. Their training pipeline combined Supervised Fine-Tuning (SFT) enriched

[10] https://openai.com
[11] https://www.anthropic.com
[12] https://deepmind.google
[13] https://www.deepseek.com
[14] https://llama.meta.com
[15] https://github.com/QwenLM/Qwen3

with Chain-of-Thought (CoT) (Wei et al., 2023) data, followed by Reinforcement Learning using Group Relative Policy Optimization (GRPO) (Shao et al., 2024). The model was trained on a diverse composite dataset, encompassing MaCoCu-tr-en (Bañón et al., 2022), WMT (Dale et al., 2025), and SIGTURK development sets.

KU-RAG. This system employs different prompting and RAG strategies across tasks, primarily leveraging Llama 4 Scout as the underlying LLM. Subtask 1 uses a prompt-based approach where contextual examples and domain-specific guidelines are provided to help the model identify technical terms and calculate exact character positions, with emphasis on capturing Turkish suffixes comprehensively (e.g., "motorlarda" not "motor"). Subtask 2 integrates expert hints directly into the prompt alongside the terms to be corrected, with strict enforcement rules ("ALWAYS use the expert hint as the base") and explicit suffix preservation instructions to minimize hint-ignored errors. Subtask 3 implements a full modular pipeline using Mistral Large[16] (1) n-gram-based term spotting; (2) fuzzy glossary retrieval from terimler.org file; (3) optional embedding-based alignment; (4) LLM post-editing with hard constraints; (5) Output Validator which is a rule-based Python module enforcing minimum glossary coverage (50%) and format compliance; and (6) Morphology Handler, which is a hybrid system combining rule-based Turkish vowel harmony and suffix patterns with LLM fallback for complex inflections. The validator and morphology handler work sequentially: validation checks constraint satisfaction, then morphology repair attempts suffix corrections before triggering a potential second LLM pass (max 1 repair, temperature=0.1). All tasks process sentences individually with full paragraph context.

KU-CoT. This system employs a multistep CoT reasoning across all tasks, guiding Llama 4 Scout LLM through structured decision steps before generating outputs. Subtask 1 uses a 5-step reasoning process: (1) verify domain-specificity; (2) check specialized scientific meaning; (3) identify Turkish equivalent with ALL suffixes; (4) calculate exact character positions by counting from position 0; (5) verify position accuracy. Subtask 2 follows a 4-step reasoning framework: (1) examine the hint to identify the correct base term; (2) analyze the current translation to identify ALL suffixes (case markers, plurals, adjectives); (3) apply those EXACT suffixes to the hint; (4) verify grammatical context fit. Subtask 3 uses a 4-step reasoning process: (1) identify English technical terms requiring translation; (2) check glossary relevance; (3) apply only high-confidence corrections ([exact] or [fuzzy 95%+]); (4) preserve all non-technical content.

Co-Text. This team uses prompts with conditional context rendering, allowing a single template to support both sentence- and paragraph-level modes. Development involved iterative refinement on the development split via error analysis, which identified suffix incompleteness, hint-ignored errors, and cross-output inconsistency. In response, the team added explicit critical rules, CoT reasoning, and consistency enforcement. Batching Strategy: Sentences from the same paragraph are grouped and processed together in a single LLM call. This approach provides richer contextual information for domain disambiguation (e.g., determining whether "field" refers to physics or agriculture), enables explicit cross-sentence consistency constraints (the same English term must use identical Turkish base forms across all sentences), and reduces API overhead while maintaining coherent terminological choices within discourse units. The team evaluates batched (paragraph-level) versus non-batched (sentence-only) processing using two backbone LLMs, namely as Mistral Large and Llama 4 Scout, to assess the consistency-efficiency tradeoff across different model architectures across the 3 subtasks.

6 Results and Analysis

Table 2 summarizes the performance of both baseline models and participant submissions across the three subtasks.

Baseline Dominance vs. Specialization. Among the zero-shot baselines, GPT-5.2 consistently outperforms other models, achieving the highest F1 (0.87) in detection and BLEU (64.82) in post-editing. However, in Subtask 3, the specialized **New Mind AI** system—finetuning a much smaller Qwen3-4B model—surpassed GPT-5.2 significantly (+5.4 BLEU). This demonstrates that domain adaptation via SFT and RL can allow compact models to outperform massive generalist models in terminology-heavy tasks.

The Difficulty of Explicit Correction. Subtask 2 (Term Correction with Expert Input) highlights a

[16] https://mistral.ai/

Model/System	Term Detection P	R	F1	Term Correction EM	End-to-End Post-Edit chrF	BLEU
Baselines						
GPT-5.2	**0.82**	**0.92**	**0.87**	0.40	83.97	64.82
Claude Sonnet 4.5	0.71	0.81	0.75	0.40	80.43	57.93
Gemini 3 Flash	0.56	0.80	0.66	0.30	74.59	52.82
DeepSeek v3.2	0.51	0.70	0.59	0.38	78.83	55.18
Llama 4 Scout	0.42	0.51	0.46	0.24	81.70	62.59
Participant Systems						
New Mind AI (Qwen3-4B)	–	–	–	–	**85.79**	**70.27**
Co-Text (Llama 4 Scout)	0.42	0.49	0.46	0.30	82.37	62.53
+ *Paragraph Context*	0.42	0.44	0.43	0.24	85.64	69.42
(Mistral Large)	0.45	0.71	0.55	0.34	76.67	49.33
+ *Paragraph Context*	0.44	0.70	0.54	0.34	77.68	50.39
Koç-CoT (Llama 4 Scout)	0.42	0.50	0.46	0.33	79.77	58.27
KU-RAG (Llama 4 Scout)	0.41	0.50	0.45	0.27	66.78	36.38
Human	–	–	0.83	**0.60**	–	–

Table 2: Results for Term Detection, Correction, and Post-Edit tasks. Participant systems are grouped by team. Best system performance in each column is bolded.

significant gap between models and human performance. While the best models achieve 0.40 Exact Match, the human expert score stands at 0.60. Error analysis reveals that models mostly revert to familiar terminology despite explicit expert hints.

Pipeline Architecture vs. Reasoning. A direct comparison between **KU-RAG** and **Koç-CoT** offers a compelling insight, as both utilized the same base model (Llama 4-Scout). Koç-CoT, employing a Chain-of-Thought (CoT) approach, achieved 58.27 BLEU, whereas KU-RAG's modular RAG pipeline dropped to 36.38 BLEU. This indicates that imposing retrieved terms as "hard constraints" disrupts fluency, whereas allowing the model to "reason" about glossary usage enables more natural integration.

Impact of Local Context. The experiments by **Co-Text** reveal a nuanced role for paragraph-level context. Grouping sentences yielded a substantial performance boost for Llama 4 Scout in the End-to-End Post-Editing task, raising BLEU scores from 62.53 to **69.42**. However, this broader context was detrimental to Llama 4's rigid Term Correction (dropping from 0.30 to 0.24 EM), an effect not observed in the Mistral Large configurations, where performance remained largely stagnant across all metrics despite the additional context.

7 Conclusion

We presented the SIGTURK 2026 Shared Task on Terminology-Aware Machine Translation, introducing the first terminology-rich parallel corpus for Turkish scientific domains with 10,157 expert-validated term links. Through a rigorous annotation pipeline, 43 trained annotators achieved substantial agreement (Fleiss $\kappa \approx 0.71$), demonstrating that high-quality specialized annotation is feasible at scale. Our comparative evaluation of state-of-the-art baselines and participant systems yielded three critical insights. First, **fine-tuning model scale**; a small model (Qwen 4B) trained on relevant data significantly outperformed massive generalist models like GPT-5.2 in end-to-end post-editing (+5.4 BLEU). Second, **reasoning outperforms rigid constraints**; systems employing Chain-of-Thought prompting to integrate terminology achieved far better fluency than modular RAG pipelines that treated terms as hard constraints. Finally, the persistent gap between automated systems and human experts in the **Term Correction** subtask (0.40 vs 0.60 Exact Match) highlights a critical limitation: models persistently revert to familiar terminology despite explicit expert hints, with the majority of errors stemming from preference for original terms over provided corrections. By openly releasing all data, code, and prompts, we provide the community with a robust benchmark to address these open challenges in terminology-aware translation.

Acknowledgments

This research is supported by the Wikimedia Foundation Research Fund (Grant No. G-RS-2402-15231). We thank Zafer Batık and Başak Tosun of the Wikimedia Community User Group Turkey for introductions to the Turkish Wikipedia community and assistance with our inquiries regarding the Wikimedia Foundation and community; Kızıl of the Wikipedia Turkey Translators Group for connecting us with translators and demonstrating the translation workflow within Turkish Wikipedia; Prof. Bülent Sankur of terimler.org for insights on tech-

nical translations and for facilitating connections with academics who contributed to terminology decisions; and Gizem Ekiz for invaluable help organizing project events and coordinating communication among academics and Wikipedians.

References

Md Mahfuz Ibn Alam, Ivana Kvapilíková, Antonios Anastasopoulos, Laurent Besacier, Georgiana Dinu, Marcello Federico, Matthias Gallé, Kweonwoo Jung, Philipp Koehn, and Vassilina Nikoulina. 2021. Findings of the WMT shared task on machine translation using terminologies. In *Proceedings of the Sixth Conference on Machine Translation*, pages 652–663, Online. Association for Computational Linguistics.

Marta Bañón, Miquel Esplà-Gomis, Mikel L Forcada, Cristian García-Romero, Taja Kuzman, Nikola Ljubešić, Rik Van Noord, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Peter Rupnik, and 1 others. 2022. Macocu: Massive collection and curation of monolingual and bilingual data: focus on under-resourced languages. In *23rd Annual Conference of the European Association for Machine Translation, EAMT 2022*, pages 303–304. European Association for Machine Translation.

J. Cohen. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37.

David Dale, Laurie Burchell, Jean Maillard, Idris Abdulmumin, Antonios Anastasopoulos, Isaac Caswell, and Philipp Koehn. 2025. Findings of the WMT 2025 shared task of the open language data initiative. In *Proceedings of the Tenth Conference on Machine Translation*, pages 495–502, Suzhou, China. Association for Computational Linguistics.

Gokhan Dogru. 2021. *Terminological Quality Evaluation in Turkish to English Corpus-Based Machine Translation in Medical Domain*. Ph.D. thesis.

Joseph L. Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76:378–382.

A. Gebeşçe, G. Şahin, and E. U. Amasya. 2025. TWiST: Turkish-English Wikipedia & Thesis STEM Terminology Dataset.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Kirill Semenov, Xu Huang, Vilém Zouhar, Nathaniel Berger, Dawei Zhu, Arturo Oncevay, and Pinzhen Chen. 2025. Findings of the WMT25 Terminology Translation Task: Terminology is Useful Especially for Good MTs. In *Proceedings of the Tenth Conference on Machine Translation*, Suzhou, China. Association for Computational Linguistics.

Kirill Semenov, Vilém Zouhar, Tom Kocmi, Dongdong Zhang, Wangchunshu Zhou, and Yuchen Eleanor Jiang. 2023. Findings of the WMT 2023 shared task on machine translation with terminologies. In *Proceedings of the Eighth Conference on Machine Translation*, pages 663–671, Singapore. Association for Computational Linguistics.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.

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

Huaao Zhang, Qiang Wang, Bo Qin, Zelin Shi, Haibo Wang, and Ming Chen. 2023. Understanding and improving the robustness of terminology constraints in neural machine translation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6029–6042, Toronto, Canada. Association for Computational Linguistics.

A Annotation Special Casese

Special Case	Description
Synonyms	When a term has synonyms listed on terimler.org, all valid translations are marked as CORRECT_TRANSLATION.
Terms connected with "and"	For terms connected by "and" (e.g., "dependent and independent variables"), each term is labeled individually, and relationships are created.
Abbreviations	When terms involve abbreviations, the full form is searched on terimler.org, and the metadata is updated. Correct translations are labeled as CORRECT_TRANSLATION, while incorrect ones are labeled WRONG_TRANSLATION.
Additional terms in Turkish	If Turkish sentences contain additional synonymous terms, relationships are created among all terms.
Latin plurals	For Latin-origin terms, such as "nucleus" and "nuclei," the singular form is searched on terimler.org and labeled accordingly.
Multi-word terms	Multi-word terms (e.g., "linear regression") are annotated as a single unit. Shorter sub-terms are not labeled separately.
Terms containing suffixes	Terms with suffixes are annotated as a whole (e.g., "conductivity"). If the suffix is mistranslated, corrections are added to the metadata.
A + term structures	Structures like "a conductive material" are annotated entirely as a single term.
Repeated terms	If terms repeat within a sentence, all occurrences are annotated.
Missing/incorrect entries on terimler.org	Various cases, including incorrect meanings, potential errors, spelling mistakes, and missing synonyms, are handled with appropriate labels (WRONG, UNCERTAIN, TYPO, SYNONYM). Comments are added to explain each situation.

Table 3: Special Cases in Annotation

B Prompts

B.1 Term Detection Prompt

You are a technical term detection expert. Your task is to identify technical/scientific terms in English sentences from scientific texts.

Task: Identify all technical terms in the given English sentence. A technical term is a specialized word or phrase used in a specific scientific domain.

Guidelines:

- Detect single words, multi-word terms, and meaningful nested terms.
- Include domain-specific concepts, processes, and specialized terminology.
- Terms are typically nouns, noun phrases, or domain-specific adjectives/verbs.

Example - what to include:
"...conformal field theory correspondence, sometimes called maldacena duality..."

- ✓"field theory" (domain concept) ✓"maldacena duality" (named theory)
- ✓"correspondence" (specific meaning) ✓"conjectured" (specific usage)

Input Data:
Source Paragraph: `{{ source_paragraph }}`
Source Sentence: `{{ source_sentence }}`
Turkish Translation: `{{ target_sentence }}`

Output Format:
Return a JSON array of objects. Each object must include:

- `en`, `en_start`, `en_end`: The English term and its 0-based character offsets (end is exclusive).
- `tr`, `tr_start`, `tr_end`: The corresponding Turkish term and its offsets.

Example Output:

```
[
  { "en": "induction motors", "en_start": 20, "en_end": 36,
    "tr": "indüksiyon motorlarda", "tr_start": 15, "tr_end": 36 }
]
```

Now, identify the technical terms in the source sentence above and return the JSON array:

B.2 Term Correction Prompt

You are an expert in translating technical/scientific terminology from English to Turkish. Your task is to correct the Turkish translations of specific technical terms based on expert-provided hints.

Context: `{{ source_paragraph }}` (if available)

Source (English): `{{ source_sentence }}`
Target (Turkish): `{{ target_sentence }}`

Terms to Correct: `{{ term_pairs }}`
(List of objects containing: English term, indices, current Turkish translation, and expert hint).

Guidelines:

1. Use the expert hint as a base; apply appropriate Turkish suffixes to match the sentence context.
2. If hint is "no-hint", provide the best technical translation.
3. Preserve morphological agreement (case, possessive, plural suffixes).

Examples:

- *Suffix Handling:* Context "...son çıkarsama hesaplamalarını...". Hint: "çıkarım".
  Correction: "çıkarım hesaplamalarını" (preserves accusative -ını).
- *No-Hint:* Context "Anything stored is data". Current: "Depolanacak" (future).
  Correction: "depolanmış" (past participle, matches meaning).

Output Format:
Return a JSON array. Each object must preserve ALL input fields (`en`, `tr`, indices, `hint`) and add a **"correction"** field.

```
[
  {
    "en": "port", ... "tr": "bağlantı", ... "hint": "bağlantı noktası",
    "correction": "bağlantı noktası"
  },
  {
    "en": "tcp three-way handshake", ... "tr": "tcp 3 yollu el sıkışmasını",
    "hint": "üç yönlü tokalaşma",
    "correction": "tcp 3 yönlü tokalaşmasını"
  }
]
```

Critical:

- Return ONLY the JSON array.
- Corrections must be fluent Turkish with proper grammatical suffixes.
- Use the hint as guidance but ensure grammatical correctness.

Now provide the corrections for the terms listed above:

B.3 End-to-End Post-Editing Prompt

You are an expert post-editor for English-to-Turkish technical/scientific translations. Your task is to post-edit a Turkish translation by correcting technical terminology while preserving the overall meaning and fluency.

```
{% if source_paragraph %}
```
Source Paragraph (English): `{{ source_paragraph }}`
```
{% endif %}
```
Source Sentence (English): `{{ source_sentence }}`

```
{% if target_paragraph %}
```
Target Paragraph (Turkish): `{{ target_paragraph }}`
```
{% endif %}
```
Target Sentence (Turkish) - TO BE POST-EDITED:
`{{ target_sentence }}`

```
{% if terminology_dict %}
```
Available Terminology Reference:
The following technical terms and their Turkish equivalents are available from terimler.org:
`{{ terminology_dict }}`
`{% endif %}`

Your Task: Post-edit the Turkish target sentence to:

1. Correct any technical/scientific term translations.
2. Use domain-appropriate Turkish terminology.
3. Ensure grammatical correctness, natural flow, and appropriate suffixes.
4. Preserve the original meaning from the English source.

Guidelines:

- Focus ONLY on technical terminology corrections.
- Keep sentence structure, word order, and non-technical words unchanged.
- Apply minimal edits - only fix terminology, not style or grammar unrelated to terms.

Example (minimal editing):

- **Original**: "Polarizasyon dağılımı, etkileşimin parametrik uzayında ve sıçrayış parametresinde iki farklı çizginin kümülatifi alınarak yeniden yapılandırıldı."
- **Changes**: "çizginin" $\rightarrow$ "yolağın"; "kümülatifi" $\rightarrow$ "kümülantı"
- **Output**: "Polarizasyon dağılımı, etkileşimin parametrik uzayında ve sıçrayış parametresinde iki farklı yolağın kümülantı alınarak yeniden yapılandırıldı."

Output Format:
Return ONLY the post-edited Turkish sentence. Do not include explanations, metadata, or the English source. Just return the corrected Turkish sentence as plain text.
Now post-edit the target sentence above: