machine learning & knowledge extraction

MDPI

Article

Improving Multilingual IT Incident Text Translation Using a Two-Stage Cascaded NMT Model Under Air-Gap Conditions

Roman Jevsejev * and Dalius Mažeika

Department of Information Systems, Vilnius Gediminas Technical University, Saulėtekio al. 11, LT-10223 Vilnius, Lithuania; dalius.mazeika@vilniustech.lt

* Correspondence: roman.jevsejev@vilniustech.lt

Abstract

Information technology service management (ITSM) systems generate large volumes of unstructured incident descriptions. They frequently include multilingual content, code-switching, informal language, and domain-specific terminology. These characteristics make automated text processing substantially more complicated and limit the applicability of conventional machine translation solutions, particularly in environments subject to strict data privacy and air-gap constraints. This paper presents a system-level reproducibility study of a deterministic two-stage cascaded neural machine translation (NMT) pipeline for normalizing multilingual IT incident text in resource-constrained, air-gapped environments. The study evaluates a sequential RU→EN and LT→EN translation strategy specifically selected to bypass unreliable language identification, enabling stable processing of code-switched incident descriptions. A system-level processing pipeline, which includes text normalization, segmentation, deduplication, adaptive batching, and language-aware data flow optimization, is analyzed to assess its impact on reducing redundant inference operations. The methodology is evaluated on a real-world ITSM dataset comprising 84,285 incident records. An incremental experimental design is used to isolate the specific contributions of computational and data-flow optimizations. Translation quality is assessed using BLEU and COMET metrics against expert reference translations produced via a primary translation and subsequent cross-verification by a second domain expert to ensure linguistic and technical consistency. The results indicate that a cascaded NMT architecture combined with systematic data-flow optimization provides a reproducible and privacy-preserving framework for multilingual IT incident text normalization, effectively supporting downstream analytical tasks in constrained operational ITSM environments.

Keywords: neural machine translation; cascaded translation; multilingual text normalization; IT incident management; code-switching; air-gap NMT pipeline; air-gapped systems; translation quality evaluation

Academic Editor: Feiping Nie

Received: 24 May 2026
Revised: 30 June 2026
Accepted: 1 July 2026
Published: 3 July 2026

1. Introduction

Information technology service management and support systems generate large volumes of unstructured textual data on a daily basis, including incident records and service requests that document service disruptions, anomalies in technical component behavior, user-reported failures, and their resolution processes in multilingual environments (specifically Lithuanian, Russian, and English). Such records are often characterized by cross-lingual code-switching, informal expression, highly specialized technical jargon, and fragmentary syntax. The combination of these characteristics creates significant challenges

for automated downstream analysis. For example, incident classification and root cause identification both require semantically consistent monolingual data representations to maintain analytical accuracy. In IT environments where Lithuanian, Russian, and English are used in parallel, incident descriptions frequently combine words or sentences from all three languages within a single text. Such trilingual segment mixtures disrupt both language identification (LID) systems [1] and standard neural machine translation (NMT) architectures. As noted in multilingual NMT (MNMT) research, models that are typically trained on homogeneous parallel corpora [2,3] perform poorly on noisy, domain-specific code-switched data. Although NMT models have reached a high level of development, empirical studies in cascaded and multilingual translation [4,5] indicate that accuracy for code-switched sentences degrades substantially as a result of unclear language boundaries, inconsistent sentence structure, or a lack of contextual information.

Moreover, real-world companies operate under strict data privacy and regulatory requirements. As a result, all sensitive technical, operational, or infrastructure-related information in IT incident descriptions (including internal system identifiers, configuration details, user data, and failure patterns) is often subject to internal security policies, contractual confidentiality obligations, and regulatory frameworks, which prohibit its transmission to external or third-party cloud-based services. As a result, the use of commercial machine translation APIs and cloud-hosted large language models becomes infeasible. These constraints impose strict operational boundaries, necessitating an investigation into fully local execution, deterministic processing, and result reproducibility within resource-constrained, CPU-only environments. Consequently, this study addresses the need for air-gapped, lightweight, and auditable translation architectures that can operate without external connectivity while guaranteeing data confidentiality and system-level reproducibility.

Despite rapid advances in NMT and research on code-switching [3,5], existing studies predominantly focus on bilingual pairs or large multilingual models [6,7] operating under unconstrained computational conditions. Although LLMs can process code-switched inputs [8], their applicability in air-gapped environments is limited by system-level reproducibility concerns, non-deterministic behavior, and prohibitive computational requirements for CPU-only execution.

Moreover, systematic reproducibility studies addressing the deterministic translation of trilingual (LT/RU/EN) IT incident texts under strict air-gap constraints remain scarce. In this context, this study investigates a two-stage cascaded air-gapped NMT architecture for processing and normalizing real-world ITSM incident texts. The pipeline operates in two sequential stages: first, Russian-language segments are translated into English (RU→EN), followed by the translation of the remaining Lithuanian fragments (LT→EN). This sequential process is designed to bypass unreliable language identification, decompose heterogeneous sentences into more interpretable linguistic components, and maintain deterministic translation behavior, which is essential for reproducible IT operations.

To address this gap, the study is built around the following research objectives:

(1) To investigate the suitability of a deterministic two-stage cascaded NMT approach for multilingual IT incident text normalization in air-gapped environments. The study aims to examine whether a fully local two-stage RU→EN and LT→EN cascaded neural machine translation pipeline can effectively process real-world ITSM incident texts characterized by code-switching and unstructured syntax while operating without explicit language identification, cloud services, or GPU acceleration.

(2) To analyze the baseline behavior of a CPU-based cascaded NMT pipeline on real operational ITSM data. This research establishes a non-optimized baseline in order to assess translation stability, computational cost, and processing characteristics when applying standard neural machine translation models to large-scale, noisy incident descriptions.

(3) To evaluate the impact of model-level computational optimization through dynamic quantization. The study investigates how applying dynamic quantization to the baseline pipeline affects inference efficiency and translation quality while the data flow and translation logic remain unchanged.

(4) To examine the effect of data-flow restructuring on translation efficiency and quality. The research explores whether pipeline-level optimizations, including length-aware batching and language-aware routing, can reduce redundant inference operations and padding overhead beyond what is achievable through model-level optimization alone.

(5) To compare incremental pipeline configurations using a controlled experimental design. The study applies a three-stage experimental framework comprising a baseline, a quantized baseline, and a data-flow-optimized pipeline in order to enable causal analysis of how individual design decisions influence performance and translation quality.

Based on these objectives, the study is led by the following research questions:

RQ1: Can a deterministic two-stage RU→EN and LT→EN cascaded NMT pipeline effectively normalize multilingual, code-switched IT incident descriptions under air-gapped and CPU-only execution constraints?

RQ2: What is the baseline translation quality and computational behavior of the proposed cascaded NMT pipeline when applied to real-world LT/RU/EN ITSM incident data without any additional optimization?

RQ3: How does dynamic quantization affect inference efficiency and translation quality in the proposed air-gapped cascaded NMT pipeline?

RQ4: To what extent do data-flow optimizations, including segmentation, deduplication, length-aware batching, and language-aware routing, reduce computational workload while preserving translation quality?

RQ5: Which experimental configuration provides the most suitable balance between translation quality, execution time, throughput, and reproducibility for practical ITSM use under constrained infrastructure conditions?

The novelty of this study is not in suggesting a new neural machine translation architecture; instead, the contribution is a system-level, deterministic, air-gapped implementation and empirical evaluation of a cascaded RU→EN and LT→EN NMT pipeline for noisy real-world ITSM incident data. The proposed approach is partially aligned with prior work on open-source machine translation, code-switched machine translation, and multilingual NMT, but it differs from previous studies by integrating normalization, segmentation, deduplication, RAM-adaptive batching, dynamic quantization, language-aware routing, and expert-based quality evaluation into a single reproducible workflow suitable for privacy-constrained operational ITSM environments. As a result, the methodological approach is partially aligned with previous studies in its use of established NMT models and standard translation quality metrics, but differs in its operational integration, deterministic execution design, air-gapped deployment constraints, and empirical evaluation on real-world multilingual IT incident data.

2. Related Works

The direction of research on machine translation has evolved from deterministic rule-based linguistic systems and statistical phrase-based translation methods toward neural sequence-to-sequence architectures and transformer models, which enable modeling of context, longer sequences, and mixed syntax. This evolution has significantly expanded the possibilities for processing multilingual and code-switched language data. Open-source initiatives such as OPUS-MT have laid the foundation for widely accessible neural translation systems. Tiedemann and Thottingal [9] emphasized the potential of developing open translation services based on the integration of multilingual corpora and easily adaptable

architectures. Subsequent work by the same authors [10] expanded the OPUS-MT ecosystem and demonstrated that open models can achieve competitive quality, attaining high BLEU, chrF, and COMET scores even for low-resource language pairs and, in some cases, outperforming systems evaluated in WMT benchmarks. The rapidly growing need to process code-switching text has stimulated research related to the effectiveness of machine translation for mixed-language data. Khatri et al. [8] analyzed the capabilities of large language models for translating code-switched content and showed that standard NMT models, which are designed for specific language pairs and are not adapted to mixed RU–LT–EN syntax, often fail to correctly interpret multiple language segments within a single sentence. A similar issue was observed by Nguyen et al. [11], who, while examining Vietnamese→English code-switching data, demonstrated that model performance significantly depends on data cleaning and the accuracy of language identification. The authors also highlighted a research gap in processing mixed-script sequences where LID remains unreliable.

Language identification in mixed texts remains a critical step prior to any translation process. Hidayatullah et al. [1] presented a systematic review of language identification for code-switched data and emphasized methodological challenges arising from uneven resource distribution across languages. The authors showed that hybrid systems combining symbolic and neural methods typically achieve higher accuracy. Meanwhile, Zhang et al. [12] proposed a selective online data augmentation methodology aimed at improving the performance of multilingual translation models. From a similar perspective, Licht et al. [13] validated the applicability of open-source translation models for quantitative text analysis. It was demonstrated that properly prepared models can substitute commercial alternatives without substantial quality loss. The specifics of code-switched social media texts were further analyzed by Srivastava and Singh [14], who developed the Hinglish corpus PHINC for mixed-language translation and NLP tasks. In a task organized by CALCS, Chen et al. [4] highlighted the complexity of translating into and out of code-switched text, demonstrating that models suffer semantic losses when language segments are tightly interwoven. These results were complemented by the GLUECoS benchmark introduced by Khanuja et al. [15], which has become one of the primary standards for evaluating NLP tasks involving code-switching. A survey conducted by Sheth et al. [5] found that the integration of large language models is fundamentally reshaping the research landscape of code-switching text, yet many architectures still inadequately process small-scale and grammatically irregular segments characteristic of real organizational data, particularly when such models must operate under the deterministic and resource-constrained requirements of air-gapped systems.

From a linguistic perspective, code-switching is closely related to language contact theory. Auer [16] interpreted code-switching as a phenomenon driven by pragmatic factors, reflecting the social and situational context of communication. Myers-Scotton [17] provided an extended analysis of bilingualism in which code-switching is viewed as a structurally predictable but semantically variable linguistic practice. Such linguistic insights are important for NMT and LID systems, as they help explain why unexpected grammatical transitions occur in mixed-language texts, potentially misleading even the most advanced neural models, and further justifying the use of deterministic cascaded stages to manage such structural variability.

The development of multilingual translation models has been driven by the creation of large-scale multilingual corpora and advances in model architectures. Fan et al. [6] proposed a multilingual model that overcomes the dominance of English in traditional NMT systems and enables translation among many low-resource languages. Masoud et al. [18] studied back-translation methods for code-switched texts and demonstrated their effective-

ness in improving translation results, particularly when the original data is fragmentary. Dabre et al. [2] presented the first systematic review of MNMT, emphasizing architectural heterogeneity, integration of different data types, and the importance of transfer learning. A later review by Dabre et al. [19] developed these findings and introduced the concept of universal multilingual architectures that can be adapted to various NMT tasks without extensive retraining. The effectiveness of transfer learning in low-resource language translation was studied by Zoph et al. [20], while Gu et al. [21] introduced meta-learning principles that allow models to adapt more rapidly to new language pairs with minimal data. These works form a theoretical foundation for streaming, resource-constrained translation processes commonly used in internal organizational systems, yet the challenge of maintaining system-level reproducibility in fully air-gapped, CPU-only environments remains. According to Cohn-Gordon et al. [3], meaning preservation is further enhanced by incorporating pragmatic reasoning into the decoding process—avoiding the collapse of semantic distinctions common in vanilla sequence-to-sequence models.

Development of new architectures and deeper models forms another significant direction of future research. Bapna et al. [22] introduced the "transparent attention" mechanism, enabling more efficient training of very deep NMT models, while Sennrich et al. [23] demonstrated that integrating monolingual data significantly improves translation quality, especially when parallel data is scarce. Kudo [7] proposed "subword regularization," which provides models with additional variability for processing incomplete or irregular word forms. Data augmentation strategies that help models better handle rare or out-of-vocabulary forms were studied by Fadaee et al. [24]. These studies showed that appropriately applied data augmentation methods can compensate for limited corpus sizes and enhance model generalization in mixed domains.

In a broader NLP context, many modern architectures have been applied beyond traditional translation tasks. Warstadt et al. [25] analyzed the ability of neural models to perform grammatical acceptability judgments, emphasizing the importance of integrating semantic and syntactic signals. Johnson et al. [26] introduced the Google MNMT system, which enables zero-shot translation between language pairs not directly seen during training. This approach was further developed by Yan et al. [27] by proposing multisource meta-transfer methods that allow more efficient processing of low-resource text queries.

Other equally important topics to consider are data privacy and security. Abadi et al. [28] demonstrated that differential privacy methods can be applied to training deep neural networks, becoming particularly relevant in contexts with high risks of data leakage. In the context of incident analysis, Zhang et al. [29] developed a corpus of medical incidents annotated for intent and factuality, revealing that precise semantic analysis is essential to avoid misinterpretation. Such principles are important for translation quality control when incident texts contain dense, domain-specific information. The BLEU metric, introduced by Papineni et al. [30], remains one of the primary standards for automatic translation quality evaluation and is widely used to assess NMT performance in both homogeneous and mixed texts. But it is important to consider that its application requires high-quality reference translations that are not always available in real organizational data. In such cases, the development of expert-verified benchmarks is required for reliable system evaluation.

Hilbig et al. [31] presented a collection of applied linguistics studies examining language contact, social multilingual communication, and the structural properties of rare languages. These aspects are particularly relevant when working with Lithuanian, which is often used in combination with other languages in complex information technology contexts. Such insights complement linguistic aspects of code-switching research required for the accurate interpretation of mixed-language texts unconstrained by strict grammatical structures, which are frequently encountered in IT incident reports.

2.1. NLP Challenges in Code-Switched IT Incident Texts

Code-switched IT incident descriptions represent a particularly challenging form of operational NLP data because they combine multilingual content, informal syntax, abbreviations, technical identifiers, and incomplete sentence structures within short and noisy textual fragments. Unlike general-domain multilingual corpora, ITSM records frequently contain mixed Lithuanian, Russian, and English expressions together with system names, error messages, device identifiers, and user-written shorthand notes. These characteristics complicate tokenization, language identification, sentence segmentation, and semantic preservation during the translation process.

Prior research confirms that code-mixed and code-switched texts create substantial difficulties for NLP systems. Hidayatullah et al. [1] showed that language identification in code-mixed text is affected by ambiguity, lexical borrowing, non-standard words, and intra-word code-mixing. Khanuja et al. [15] introduced GLUECoS as a benchmark for code-switched NLP tasks, demonstrating that code-switching affects multiple downstream tasks, including language identification, named entity recognition, sentiment analysis, and question response. Khatri et al. [8] further showed that code-switched machine translation remains challenging even for large multilingual language models. Nguyen et al. [11] also demonstrated that code-switching input introduces specific methodological difficulties for machine translation evaluation. Similarly, Chen et al. [4] emphasized the demand for dedicated machine translation resources and evaluation settings for code-switched data.

These challenges justify the use of a deterministic staged translation strategy rather than a single end-to-end multilingual translation step. A cascaded RU→EN and LT→EN pipeline allows mixed-language incident descriptions to be decomposed into more stable processing stages, reducing dependence on explicit language identification and improving auditability of the translation process. This is especially relevant for operational ITSM environments, where reproducibility and traceability are required for downstream analysis such as incident classification, root-cause analysis, SLA analytics, and retrospective service improvement.

2.2. Security and Privacy Constraints in Operational NLP

Operational NLP pipelines for IT incident data must also address strict security and privacy constraints. Incident descriptions may contain internal system identifiers, infrastructure names, configuration details, failure patterns, user-related information, and other organization-specific operational context. As a result, transferring such data to external machine translation APIs, cloud-hosted LLMs, or third-party NLP services may lead to violations of internal security rules, contractual confidentiality obligations, or data protection requirements. These constraints are especially relevant in critical or highly regulated environments, where data isolation and auditability are not optional implementation choices but mandatory design requirements.

The importance of privacy-preserving processing has been emphasized in broader research on machine learning. Abadi et al. [28] showed that machine learning datasets may contain sensitive information, and that model development must account for privacy risks. In the machine translation domain, Tiedemann and Thottingal [9] presented OPUS-MT as an open translation ecosystem intended to support transparent and secure translation services. Tiedemann et al. [10] further emphasized that reliance on commercial machine translation involves risks related to data privacy, transparency, and inclusivity. Licht et al. [13] also showed that open-source machine translation can support reproducible multilingual text analysis as an alternative to commercial translation services.

For this reason, air-gapped and fully local NLP execution becomes a methodological requirement for privacy-preserving incident text processing. When compared with cloud-

based translation services, an offline NMT pipeline provides stronger control over data locality, reproducibility, and operational auditability. However, this design also introduces computational constraints, particularly when only CPU-based infrastructure is available. Therefore, research on multilingual ITSM translation must jointly consider linguistic quality, computational efficiency, reproducibility, and information protection.

In summary, the processing of multilingual and code-switching text poses significant methodological challenges for neural machine translation models, particularly when language segments are tightly interwoven, and resources for specific languages are limited. Research indicates that successful mixed-language NMT systems must integrate linguistic principles, advanced language identification methods, MNMT architectures, data augmentation, meta-learning strategies, and privacy-preserving mechanisms. This integrated methodological foundation enables the development of more reliable systems adapted to real-world domains in which translation accuracy is critically important.

However, the implementation and reproducibility of such systems within strict air-gap and CPU-only constraints remain a secondary focus in current literature, necessitating dedicated systems-oriented studies.

Based on the analyzed references, it can be concluded that the processing of multilingual and code-switching text remains one of the most challenging problems in neural machine translation, especially when dealing with real organizational data characterized by irregular structure, fragmentary syntax, and limited linguistic resources. Studies reveal that applying a single universal model to mixed multilingual sentences often results in semantic loss, insufficient separation of language segments, and limited stability in domains that are not well represented in standard NMT training datasets. For these reasons, the scientific literature increasingly emphasizes the need for staged translation strategies, hybrid approaches, and selective data transformation techniques that help circumvent unreliable language identification in mixed texts and enable more accurate translation. In this study, we adopt a staged cascaded architecture as a deliberate design choice to ensure system-level determinism and auditability, which are key requirements for reproducibility in air-gapped IT operations that are often overlooked by end-to-end multilingual models.

Within this context, the OPUS-MT model ecosystem stands out as a suitable solution for low-resource languages, as these models are accessible, capable of operating in air-gapped environments, and provide sufficient quality for limited-domain tasks. Moreover, many studies stress the importance of privacy and data isolation in practical NLP workflows, noting that sensitive information cannot be transmitted to external translation services. Consequently, air-gapped NMT architectures become a methodological requirement rather than an optional criterion. At the same time, research shows that effective translation of mixed domains requires not merely architecturally complex models, but a consistent and reproducible pipeline of data preparation, segmentation, and quality control. This study contributes to this direction by evaluating how such a pipeline compensates for linguistic non-standardization and terminological fragmentation within the strict operational boundaries of CPU-only infrastructure.

3. Research Methodology

Based on the regularities identified in the literature [2,3], an air-gapped two-stage RU→EN and LT→EN translation architecture was selected for this study as a system-oriented and empirically grounded solution. This approach represents a deliberate design choice aimed at decomposing code-switched IT incident descriptions into deterministic processing stages and bypassing unreliable language identification to ensure system-level reproducibility [1,4] for short, mixed, and semantically fragmented texts typical of real ITSM environments.

The research methodology is designed as a consistent, multi-stage data processing and translation pipeline beginning with automated field identification and data loading, followed by intensive text normalization to stabilize the input. The core of the system is a cascaded translation engine optimized for CPU-only execution through algorithmic and structural enhancements. The final stages involve result consolidation and systematic quality assessment against an expert-verified reference. By utilizing this deterministic step-by-step pipeline, the system ensures the preservation of the semantic integrity of technical identifiers, providing more reliable input for downstream models than stochastic end-to-end alternatives. The overall workflow of the proposed cascaded translation methodology is illustrated in Figure 1.

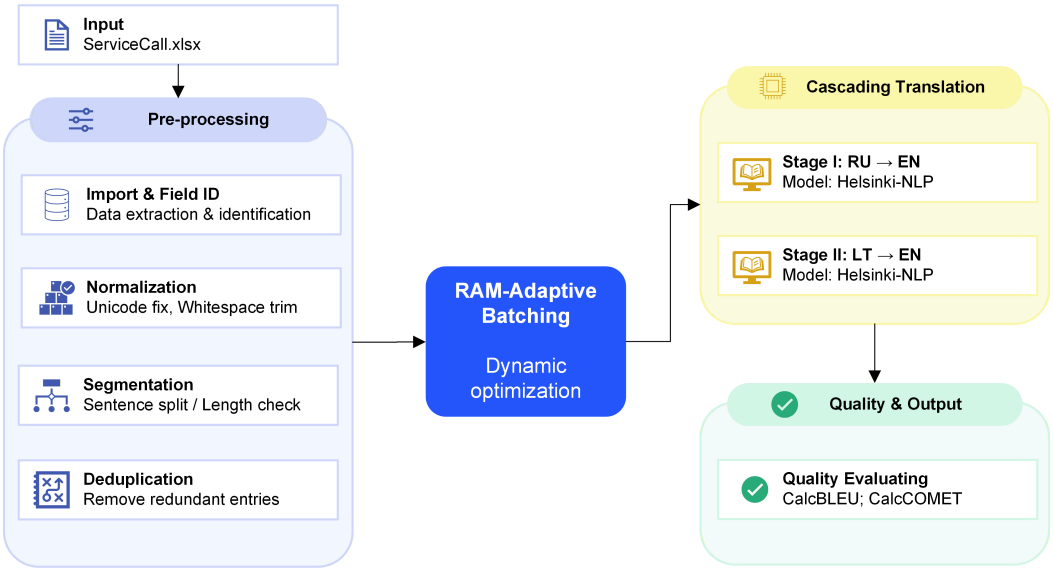
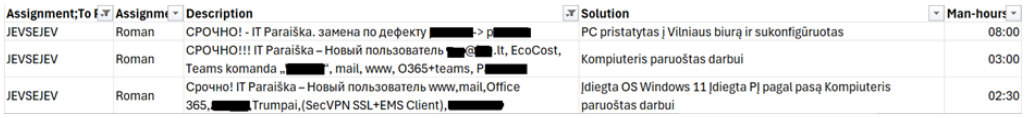


Figure 1. Overall architecture of the proposed two-stage RU→EN and LT→EN cascaded translation pipeline.

3.1. Data Import, Dataset Structure, and Field Identification

The primary data source is the ServiceCall.xlsx file exported from the organization's ITSM system. The dataset was used exclusively for methodological evaluation of multilingual incident text normalization. Personal names and assignment-related fields were excluded from the translation input and were not used for evaluating individual users, employees, or service desk personnel. The analysis focuses on textual normalization, translation quality, processing efficiency, and aggregate experimental outcomes rather than on individual-level behavioral assessment. Operationally sensitive identifiers and confidential organizational information are subject to internal data protection and security restrictions; therefore, the original dataset cannot be made publicly available. Data import is performed using auxiliary functions that read the selected worksheet and automatically verify whether the file structure is suitable for further processing. During import, empty rows are removed, data types are standardized, and the number of columns is validated, thereby ensuring stable and deterministic data loading.

Figure 2 shows an anonymized excerpt of the original ITSM incident dataset used in this study. The example illustrates the multilingual and code-switched nature of the description field, where Lithuanian, Russian, and English fragments may appear together with technical identifiers, application names, e-mail references, and informal service desk notes. This figure also shows solution and man-hours fields, which provide operational context for later aggregate analysis but are not used to identify or evaluate individual users, employees, or service desk personnel.



Assignment;To f...	Assignme...	Description	Solution	Man-hours
JEVSEJEV	Roman	СРОЧНО! - IT Paraiška. замена по дефекту [redacted] -> p[redacted]	PC pristatytas į Vilniaus biurą ir sukonfigūruotas	08:00
JEVSEJEV	Roman	СРОЧНО!!! IT Paraiška – Новый пользователь [redacted]@[redacted].lt, EcoCost, Teams komanda „[redacted]", mail, www, O365+teams, P[redacted]	Kompiuteris paruoštas darbui	03:00
JEVSEJEV	Roman	Срочно! IT Paraiška – Новый пользователь www,mail,Office 365,[redacted]Trumpai,(SecVPN SSL+EMS Client),[redacted]	Įdiegta OS Windows 11 Įdiegta PĮ pagal pasą Kompiuteris paruoštas darbui	02:30

Figure 2. Excerpt of the original ITSM incident dataset with multilingual code-switching descriptions.

The loaded dataset comprises 84,285 incident records and includes a total of 9 attributes, covering administrative, temporal, assignment-related, and textual information. Although only a limited set of columns (Description, Solution, Actual Finish) is directly used in subsequent translation stages, it is necessary to define the complete data structure, as certain fields are critical for filtering purposes (e.g., distinction between open and closed incidents), while others may be utilized in later analyses such as incident classification, SLA analysis, or assessment of personnel workload. The key fields of the ITSM incident dataset used in this study are summarized in Table 1.

Table 1. Key ITSM incident dataset fields.

Field	Description
ID	Unique incident identifier; used for record linking.
Registration date	Incident registration timestamp; used for noise analysis, durations, and SLA modeling.
Status	Incident state (Open/Closed); used for data splitting.
Actual Finish	Completion timestamp; used to calculate resolution duration.
Caller names	Client first and last name (LT); excluded from translation; anonymized.
Assignment names	Resolver names; excluded from translation; used for workload analysis.
Description	Main raw incident text (RU/LT/EN); primary input for segmentation and translation.
Solution	Resolution text; reserved for future NLP analysis.
Man-hours	Work effort metric; used for SLA and performance analysis.

The Man-hours field represents an operational work-effort value recorded in the ITSM system during incident handling. It should be interpreted as an administrative proxy rather than as a directly validated measurement of actual labor time. In this study, Man-hours is not used as a primary translation quality metric and is not used to evaluate NMT performance. Its relevance is limited to later SLA-oriented analyses, where translated incident descriptions may be linked with work effort, resolution duration, incident complexity, and service performance indicators. Because the field depends on manual reporting practices, it may contain inconsistencies and should be used cautiously.

Since ITSM data column names can be heterogeneous, multilingual, abbreviated, or subject to spelling variations, automated field identification is performed after import. A combined approach, incorporating regular expressions (regex), fuzzy matching similarity measures, and semantic keyword voting, ensures that true semantic fields are reliably identified. The identified columns are then automatically normalized into a standardized schema, which is consistently used by the translation model in subsequent stages. This solution enables robust methodology operation across heterogeneous datasets, eliminates manual variable labeling, and ensures that a structurally consistent and semantically reliable input dataset enters the translation pipeline. The incident data import and semantic field identification process is depicted in Figure 3.

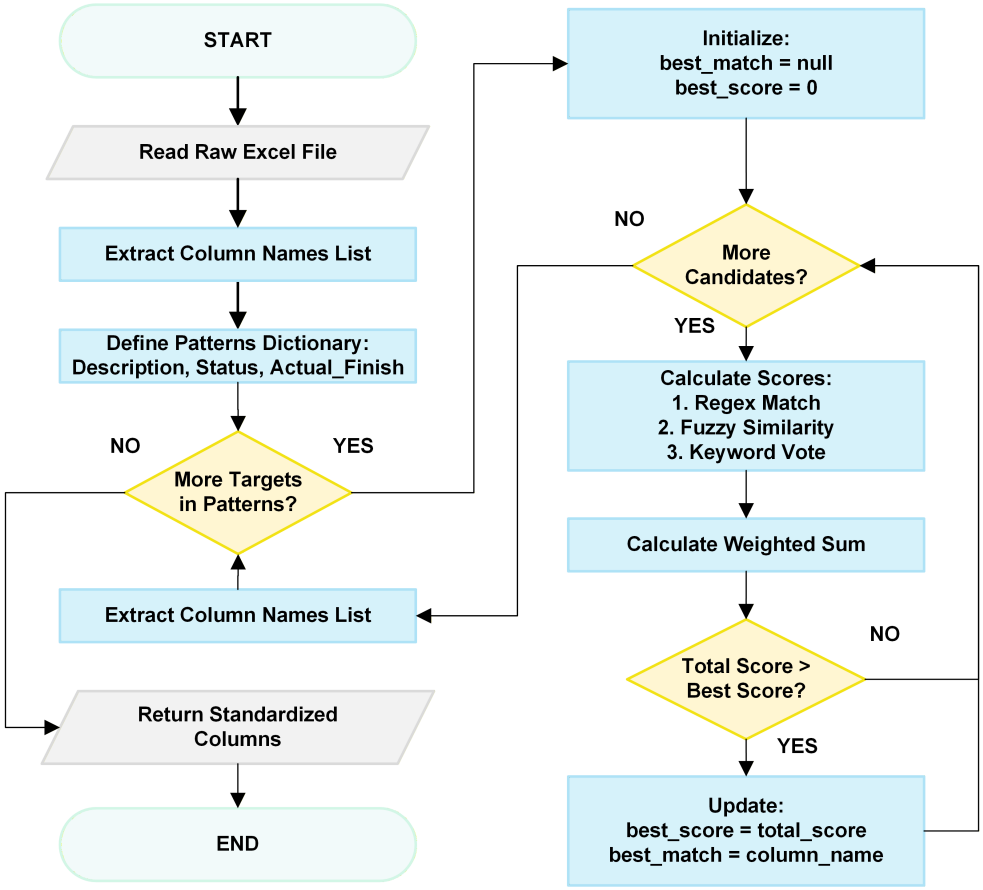


Figure 3. Data import and semantic field identification process for heterogeneous ITSM incident datasets.

3.2. Text Normalization and Segmentation

Incident descriptions contained in the description column demonstrate a high degree of structural and linguistic diversity: some are presented in a formal manner, while the majority consist of mixed-language (LT/RU/EN) fragments combined with extraneous symbols, abbreviations, and multi-line blocks. This unstructured nature complicates the operation of machine translation models, so comprehensive text normalization and segmentation are performed prior to applying neural translation. The normalization objectives are noise reduction, removal of elements that interfere with model operation, and preparation of the text for efficient translation. This stage is critical for ensuring system-level reproducibility and deterministic behavior of the NMT models under strict operational constraints. The applied text preprocessing steps, including normalization, segmentation, and deduplication, are illustrated in Figure 4.

The first step involves the removal of invisible or undesired Unicode control characters, as these can disrupt tokenization and lead to unpredictable model behavior. Next, whitespace and line structures are standardized: excessive spaces are collapsed, multiple consecutive line breaks are replaced with single ones, and CR/LF sequences are normalized into a unified format. This step eliminates inconsistencies arising from different systems, varying input practices, or copy-paste operations.

After basic normalization, text segmentation is performed using a two-level strategy. Initially, natural sentence boundaries are identified using punctuation marks and capitalization in both Latin and Cyrillic alphabets. In cases where sentence boundaries cannot be reliably detected, segmentation is performed based on a maximum allowable fragment length, in order to prevent memory overload during the translation stage. This segmentation yields smaller, semantically clearer text segments.

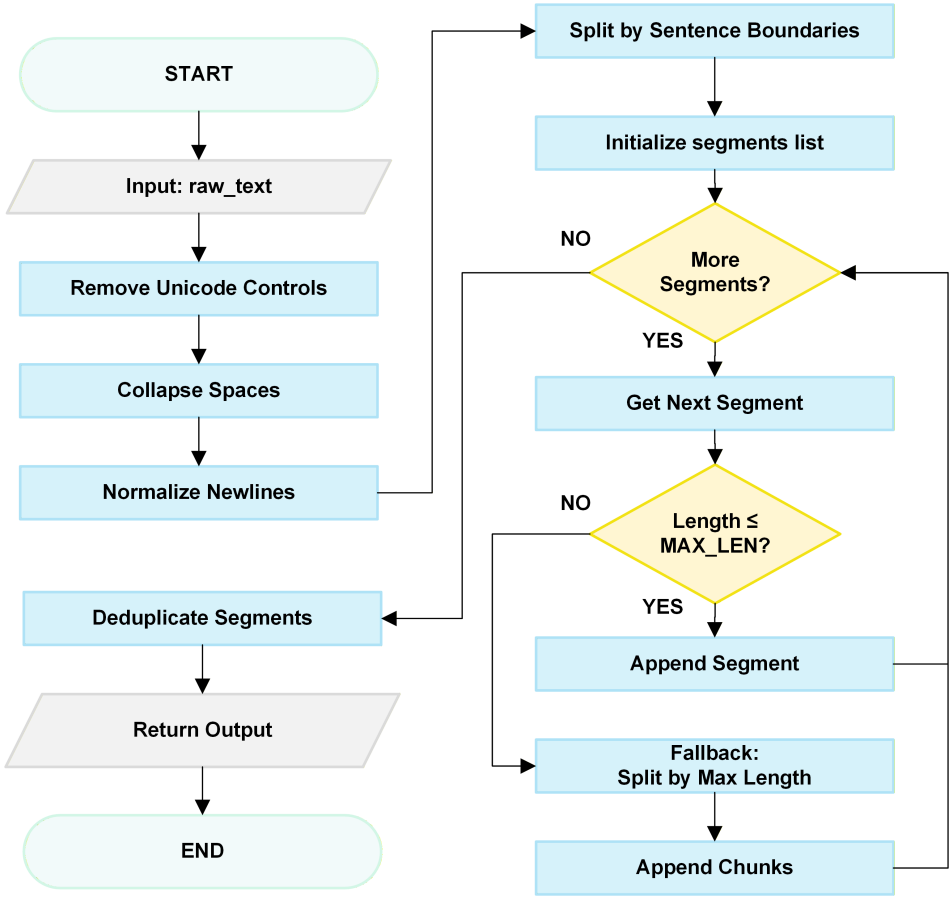


Figure 4. Text normalization, segmentation, and deduplication workflow for multilingual incident descriptions.

After segmentation is finished, deduplication is performed: identical segments are extracted only once and stored in a separate cache list. Since incident descriptions often contain recurring standard phrases (e.g., "login does not work," "please reboot," etc.), deduplication reduces the number of redundant translation operations. Finally, normalized segments are assembled into structures suitable for batching, and their lengths are calculated.

After following these steps, the original text becomes structured and semantically predictable. Segmentation ensures that long mixed-language descriptions are divided into optimally sized fragments that the model can process without loss of context. Deduplication reduces the overall translation workload, while the standardized segment structure enables efficient application of batching.

In order to evaluate the linguistic complexity of the dataset and to identify potential challenges for automated routing, a quantitative audit of the 84,285 incident records was performed. This analysis was focused on identifying specific edge cases that typically disrupt standard language identification (LID) systems.

And the results revealed that 487 records (0.5778%) contained mixed-script tokens where Cyrillic and Latin characters were combined within a single word (e.g., "server" + Cyrillic A). Additionally, 277 records (0.3286%) featured Latin-scripted Russian transliteration (e.g., "perezagruzit" or "oshibka"). These cases constitute a total "router noise" floor of 0.9064%, representing over 760 instances where a conventional script-based router would fail to initiate the required translation stage.

3.3. Adaptive Batching in a Resource-Constrained Computing Environment

OPUS-MT neural machine translation models process textual data not as individual lines but in batches, that is, groups of multiple text sequences processed within a single

model execution cycle. Such batching enables efficient utilization of CPU computational capacity and reduces overall translation time, especially when working with large-scale datasets. The RAM-adaptive batching mechanism used to optimize inference efficiency is illustrated in Figure 5.

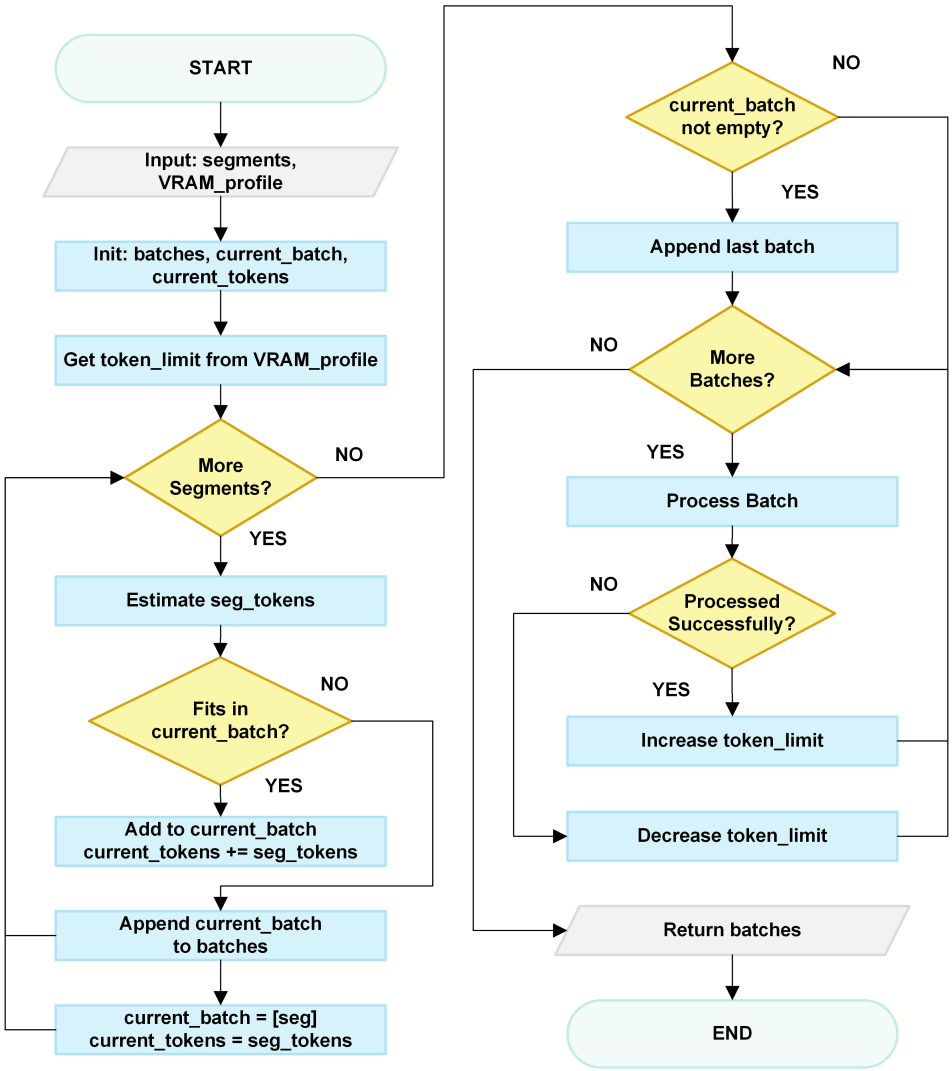


Figure 5. Resource-adaptive batching strategy for CPU-based neural machine translation execution.

Since all experiments in this study are conducted in CPU mode without GPU acceleration, batch size is constrained not by graphics memory (VRAM) but by system memory (RAM), according to processor load and execution stability criteria. For this reason, a dynamic, resource-adaptive batching strategy is employed, which automatically adjusts to actual execution conditions and allows stable model operation to be maintained without memory exhaustion or significant performance fluctuations.

For each text segment, an approximate token count is calculated prior to translation, and batches are constructed in a way that their total size does not exceed a predefined safe computational threshold. This ensures that model execution remains stable even while processing longer or structurally more complex incident descriptions. Batch size is adjusted adaptively based on the results of previous executions, gradually achieving an optimal compromise between performance and reliability. This approach makes the solution independent of specific hardware configurations and allows it to be directly applied in other computing environments, including GPU-based systems. It also ensures

both experimental stability under resource-constrained conditions and methodological portability to higher-performance infrastructures.

3.4. Cascaded Two-Stage Translation

After text normalization and segmentation, each incident description fragment is passed to a two-stage translation model, which forms the core of the methodology employed in this study. The language-aware data flow routing strategy introduced in the optimized configuration is shown in Figure 6.

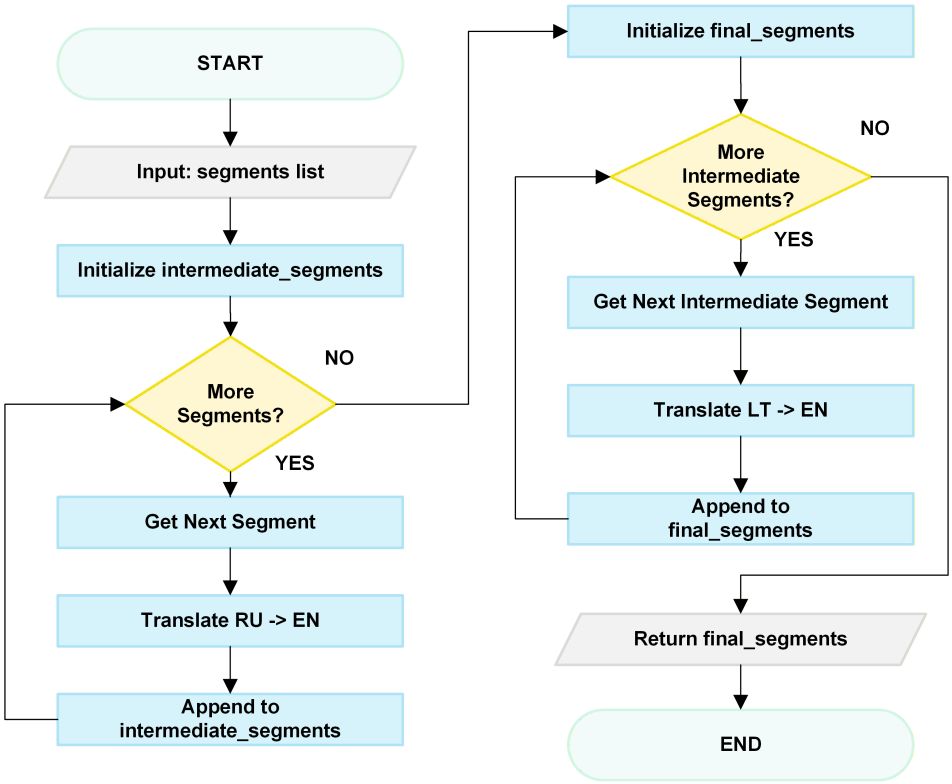


Figure 6. Logical structure of the two-stage cascaded translation process with intermediate result handling.

Initially, all segments are translated from Russian into English using the Helsinki-NLP/opus-mt-ru-en model. At this stage, Cyrillic-language fragments are eliminated, while Lithuanian-language words remain unaffected, because the model does not recognize them as Russian ones. This ensures that LT fragments are not erroneously converted and that their interpretation is postponed until the second, specialized translation stage. Since Russian-language elements represent the most frequent code-switching component in incident texts, this stage removes structural interference that would otherwise hinder the second model.

In the second stage, the resulting mixed text is translated from Lithuanian into English using the Helsinki-NLP/opus-mt-tc-big-lt-en model. At this stage, the remaining Lithuanian-language fragments are translated, while words already in English remain unchanged. Because the first stage has removed Russian fragments and the normalization and segmentation processes have ensured appropriately sized inputs for the models, the second stage operates stably and accurately, without the ambiguity characteristic of mixed RU+LT sentences.

The selection of the cascaded two-stage model is motivated by several fundamental linguistic and technological considerations. First, there is no effective air-gapped model that operates without GPU VRAM and is capable of reliably processing trilingual LT+RU+EN

sentences in a dataset characterized by grammatical disorder and syntactic ambiguity. Second, large multilingual models that could theoretically perform such translation cannot be applied in practice due to strict privacy, resource, and infrastructure constraints, as an air-gapped environment eliminates the possibility of using cloud-based services. Third, the two-stage sequence allows code-switching texts to be constructively decomposed into logical components by removing one language layer at a time, thereby reducing the risk of model errors.

3.5. Result Consolidation and Structuring

In the final stage of the methodology, all intermediate translation and quality control results are consolidated into a unified, structured dataset, which is used not only for generating the final output files but also for the systematic evaluation of neural machine translation results. At this stage, the methodology is extended beyond the mere accumulation of automated translation outputs to include their comparison with expert reference translations using COMET and BLEU metrics. The automated translation quality evaluation process based on BLEU and COMET metrics is presented in Figure 7.

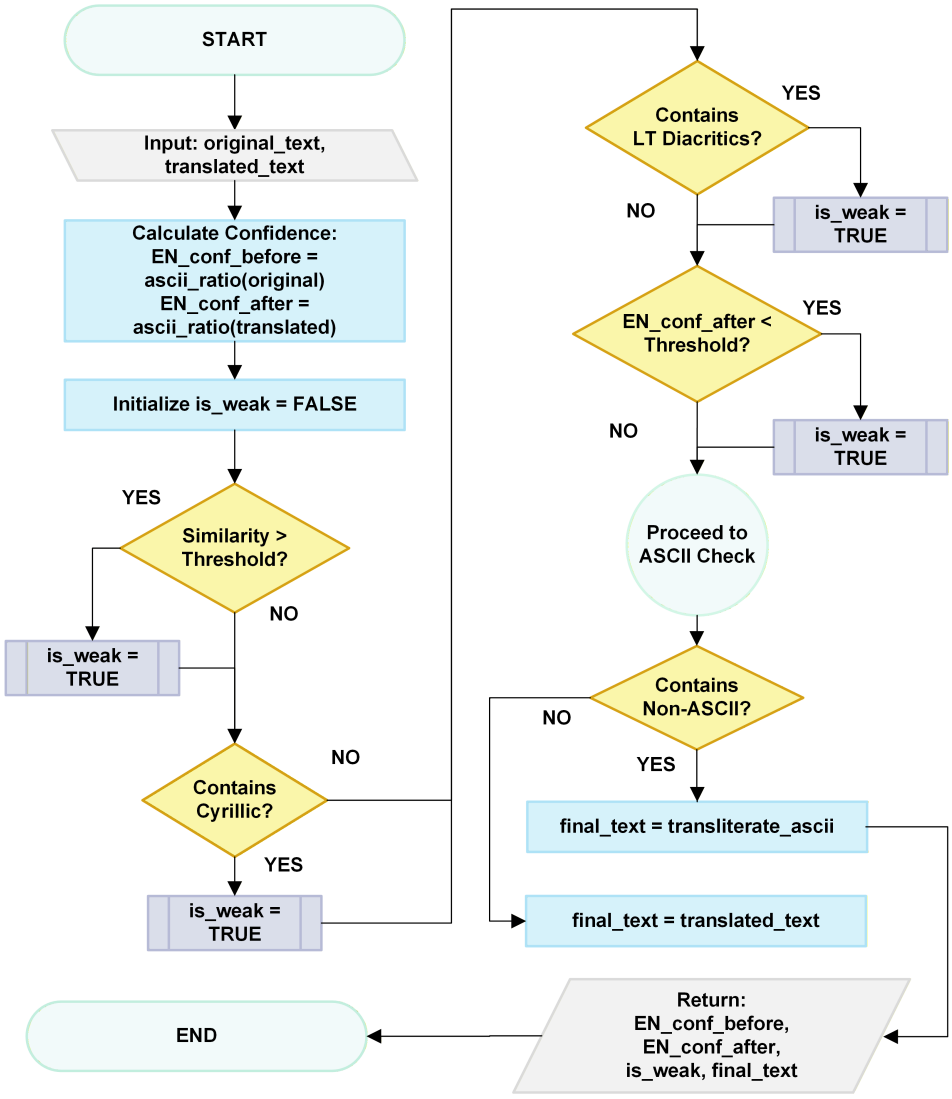


Figure 7. Translation quality control and weak-output detection logic within the cascaded pipeline.

Expert reference translations were produced for the research dataset and used as references for comparison of the results. This reference translation enables objective assessment

of NMT output quality using BLEU and COMET metrics. For each incident record, the final neural English translation and the corresponding expert translation are paired, forming a direct basis for comparison at the individual incident level.

In order to partition incidents into analytically meaningful groups, a strict classification rule is applied: an incident is considered closed only if its status is marked as "closed" and if the Actual Finish field is filled, indicating the factual completion date. All records that do not meet these criteria are automatically assigned to the open incident group. Based on this classification, two separate datasets are formed: closed incidents and open incidents, which are exported into separate files named Translated_Closed_Tickets.xlsx and Translated_Open_Tickets.xlsx. If required, a combined aggregated dataset can also be generated for global analysis. This consolidated scheme enables clear separation of analytically suitable completed records from active incidents, preserves full linkage to all original data, and provides a solid foundation for consistent downstream analysis, retrospective incident evaluation, and future supervised modeling if required.

4. Experimental Environment and Reproducibility

An incremental experimental design strategy is applied in this study, whereby each subsequent experiment does not replace a previous methodological decision but consistently extends and enriches it. This design choice is made deliberately in order not only to improve overall system performance but also to empirically identify specific practical limitations that arise in real-world IT incident translation processes. All experiments are conducted strictly in an air-gapped environment to ensure data confidentiality and deterministic system behavior. The hardware and software configuration meeting the minimum reproducibility requirements for the experiments is presented in Table 2.

Table 2. Minimal reproducibility environment (hardware and software).

Parameter	Specification
Operating system	Windows 11 Enterprise (Build 26200)
Processor (CPU)	AMD Ryzen 5 8540U (6 cores, 12 threads)
CPU frequency	3.20 GHz
Memory (RAM)	32 GB
Compute mode	CPU execution only
Exec. env.	Jupyter Notebook 7.4.4
Python version	Python 3.10.18
Package manager	Conda 24.11.3
pip version	pip 25.1

All translation experiments were executed locally without external API calls, cloud-based translation services, or network-dependent inference components. The RU→EN translation stage used the Helsinki-NLP/opus-mt-ru-en model, while the LT→EN stage used the Helsinki-NLP/opus-mt-tc-big-lt-en model. Translation quality was evaluated using BLEU and COMET metrics against expert reference translations. The experimental design was incremental: the baseline FP32 configuration was first evaluated, followed by dynamic quantization and then by data-flow optimization through length-aware batching and language-aware routing. This design enables direct comparison of individual optimization effects under identical dataset and execution conditions. The combination of a two-stage air-gapped translation architecture, an incremental experimental design strategy, and strictly defined reproducibility conditions constitutes the core methodological contribution of this study. The complete pipeline implementation and additional experimental results are provided in the Supplementary Materials. The results indicate that hybrid neural machine translation strategies, which deliberately bypass direct language identification and operate in an air-gapped environment, can deliver reliable, reproducible,

and practically applicable translation outcomes for complex code-switching IT incident data without the use of large GPU resources.

By addressing practical challenges of incident management systems—such as data privacy, deterministic execution, and constrained computational resources—the proposed methodology contributes to the advancement of applied machine translation technologies and is explicitly oriented toward real-world ITSM processes.

Each experimental stage was formulated based on the shortcomings identified during the preceding experiment, allowing all architectural and computational decisions to be clearly and rigorously justified. This section provides a detailed account of the issues identified after each experiment and explains how they motivated the introduction of subsequent experimental stages, thereby ensuring methodological transparency and traceability of decisions.

4.1. Experiment: Baseline Experimental Configuration

In the first experiment, a baseline configuration of the proposed methodology was implemented in order to validate the operation of the two-stage cascaded neural machine translation scheme on a real IT incident dataset. The experiment was conducted without any additional performance optimizations, with the aim of evaluating the initial behavior of the method in an air-gapped, CPU-based environment.

The entire research dataset, consisting of 84,285 incident records, was used in the experiment. After text normalization, segmentation, and deduplication, 58,817 unique textual fragments were obtained, constituting the actual translation workload. All fragments were sequentially passed through both translation stages: first applying the RU→EN model, followed by the LT→EN model.

No supervised model training or fine-tuning was performed in this study; therefore, no conventional training/validation split was required for model optimization. The Helsinki-NLP models were used as fixed, pre-trained NMT models under offline inference conditions. The dataset was not partitioned into training and validation subsets because the experimental objective was to evaluate a deterministic inference pipeline rather than to optimize model parameters. Deduplication was performed before translation and quality evaluation to prevent repeated textual fragments from inflating the reported results. The complete deduplicated fragment set was processed through the same deterministic pipeline, and translation quality was assessed against expert reference translations.

In this experiment, no logical separation of the data flow based on linguistic characteristics was applied—each fragment was processed by both translation stages. This design choice was made in order to capture a baseline "full-path" execution scenario.

The entire translation process was executed in a CPU environment using FP32 computational precision, thereby ensuring a conservative and easily reproducible experimental setup.

4.2. Experiment 2: Application of Quantization in the Baseline Configuration

In the second experiment, the same methodological structure, dataset, and translation flow logic as in the first experiment were retained. The objective of this stage was to evaluate the impact of dynamic quantization on the translation process without altering the overall experimental scheme.

The only methodological modification was introduced at the level of the model's computational representation: the linear layers of the neural networks were quantized from FP32 to the qint8 format using a dynamic quantization approach. The model architecture, training data, and inference logic remained unchanged.

As in the first experiment, the entire dataset of 84,285 incident records was processed, yielding the same 58,817 textual fragments after normalization and deduplication. All fragments were sequentially passed through both translation stages, without applying any linguistic filtering or flow optimization mechanisms. This ensured direct comparability of the results with the baseline configuration.

4.3. Experiment 3: Data Flow Optimization and Computational Load Reduction

The third experiment focused on optimizing the data processing logic while retaining the quantized model representation and the two-stage cascade structure applied in the previous stages. The goal of this stage was to reduce redundant computations arising from non-optimal fragment handling and a universal translation flow.

The same dataset was used—84,285 incident records—from which 58,817 unique textual fragments were obtained after normalization and deduplication. Unlike the first two experiments, fragments were not automatically passed through both translation stages.

The first applied modification was sorting fragments by length prior to batch formation (smart batching), with the aim of reducing the number of padding tokens and the associated computational overhead. This change did not affect model outputs, as only the organization of inputs was modified.

The second modification involved logical separation of the data flow based on linguistic characteristics. By introducing a Cyrillic character detection mechanism, fragments were routed only to the translation stage required for them. As a result, 42,426 fragments were processed in the RU→EN stage, while 16,391 fragments were processed in the LT→EN stage, thereby eliminating redundant double processing.

5. Results

The conducted experimental study enabled a systematic evaluation of the performance and translation quality of the two-stage cascaded neural machine translation methodology under different experimental configurations. Execution time and effective throughput were selected as primary operational efficiency metrics because the proposed pipeline targets CPU-only and air-gapped environments, where large-scale translation must be completed locally without cloud infrastructure. Execution time captures the total processing cost of a complete translation run, while throughput indicates how many incident fragments can be normalized per unit of time, directly reflecting practical deployability in ITSM settings. A comparative summary of execution time and effective throughput across all three experimental configurations is provided in Table 3.

Table 3. Comparative overview of the three experimental configurations.

Experiment	Configuration	Main Change	Data-Flow Logic	Purpose and Metrics
E1	Baseline FP32	No optimization; FP32 inference	All fragments are processed through RU→EN and LT→EN	Conservative baseline; execution time, throughput, BLEU, COMET
E2	Quantized baseline	Dynamic quantization	Same full-path processing as E1	Model-level efficiency gain; execution time, throughput, BLEU, COMET
E3	Optimized pipeline	Quantization, batching, routing, deduplication	Language-aware and length-aware processing	System-level optimization effect; execution time, throughput, BLEU, COMET

A detailed execution time and throughput summary for each experiment is provided in Table 4.

Table 4. Execution time and throughput summary across three experimental configurations.

Experiment 1: Baseline FP32, Full Cascaded Execution	
Unique texts	58,817 (from 84,285 total records)
Total execution time	8:51:27 (RU→EN: 2:51:41; LT→EN: 5:59:19)
Effective throughput	2.64 rows/s (RU→EN: 5.71 it/s; LT→EN: 2.73 it/s)
Experiment 2: Quantized (FP32→qint8), full cascaded execution	
Unique texts	58,817 (from 84,285 total records)
Total execution time	2:34:02 (RU→EN: 1:10:02; LT→EN: 1:23:13)
Effective throughput	9.12 rows/s (RU→EN: 14.00 it/s; LT→EN: 11.78 it/s)
Experiment 3: Quantized, smart batching, and language-aware routing	
Unique texts	58,817 total (RU: 42,426; LT: 16,391)
Total execution time	0:42:07 (RU→EN: 0:28:17; LT→EN: 0:13:22)
Effective throughput	33.35 rows/s (RU→EN: 25.00 it/s; LT→EN: 20.42 it/s)

5.1. Results of the Baseline Configuration

The performance results obtained in the first experiment revealed clear limitations of the baseline configuration. The average translation throughput of Russian fragments in the RU→EN stage reached 5.71 fragments per second, while the translation throughput of Lithuanian fragments in the LT→EN stage was 2.73 fragments per second. Due to this asymmetry, the second translation stage became the dominant contributor to the overall processing time.

The total execution time of the RU→EN stage was 2 h 51 min 41 s. The LT→EN stage required 5 h 59 min 19 s. Overall, the complete two-stage translation of the entire dataset of 84,285 incident records took 8 h 51 min 27 s. When considering the full processing pipeline, the effective system throughput was 2.64 incident records per second.

In terms of translation quality, the baseline configuration achieved BLEU and COMET scores of 36.93 and 79.45, respectively. These results are used in this study as reference points for subsequent experimental comparisons.

The takeaway confirmed that the baseline methodology is functionally correct and capable of stably processing large-scale, mixed-language IT incident data in a local environment. At the same time, significant practical limitations were also identified, including long overall execution time, relatively low effective throughput, and high computational load in a CPU-based environment. In addition, it was determined that routing all textual fragments through both translation stages and forming batches without prior length-based sorting resulted in redundant computations and inefficient resource utilization.

5.2. Impact of Dynamic Quantization on Performance and Quality

In the second experiment, the application of dynamic quantization—without modifying the data flow logic—resulted in a substantial improvement in execution performance. The RU→EN stage was completed in 1 h 10 min 02 s, while the LT→EN stage required 1 h 23 min 13 s. The total execution time of the two-stage translation process was reduced to 2 h 34 min 02 s, and the effective system throughput increased to 9.12 incident records per second.

These results confirmed that optimization of the model's computational representation alone can have a significant impact on practical system performance. Dynamic quantization substantially reduced memory consumption, enabled stable inference in a CPU environment, and rendered the methodology realistically applicable for organizations that either lack access to GPU infrastructure or deliberately avoid it due to cost or security requirements.

Translation quality evaluation revealed a small but quantitatively measurable decrease: the BLEU score declined from 36.93 to 35.70 (−3.33%), and the COMET score decreased from 79.45 to 78.82 (−0.79%). These findings indicate that computational representation optimization, although highly beneficial for performance, is not sufficient on its own to achieve maximum system effectiveness.

5.3. Results of Data Flow Optimization

The third experiment demonstrated the strongest overall methodological effect, as optimization was applied not to the computational representation of the neural translation models but to the logic of textual data presentation and processing. By applying length-based segment sorting and logical data flow separation according to linguistic characteristics, a substantial performance gain was achieved. The comparison of effective processing speed across the three experimental configurations is shown in Figure 8.

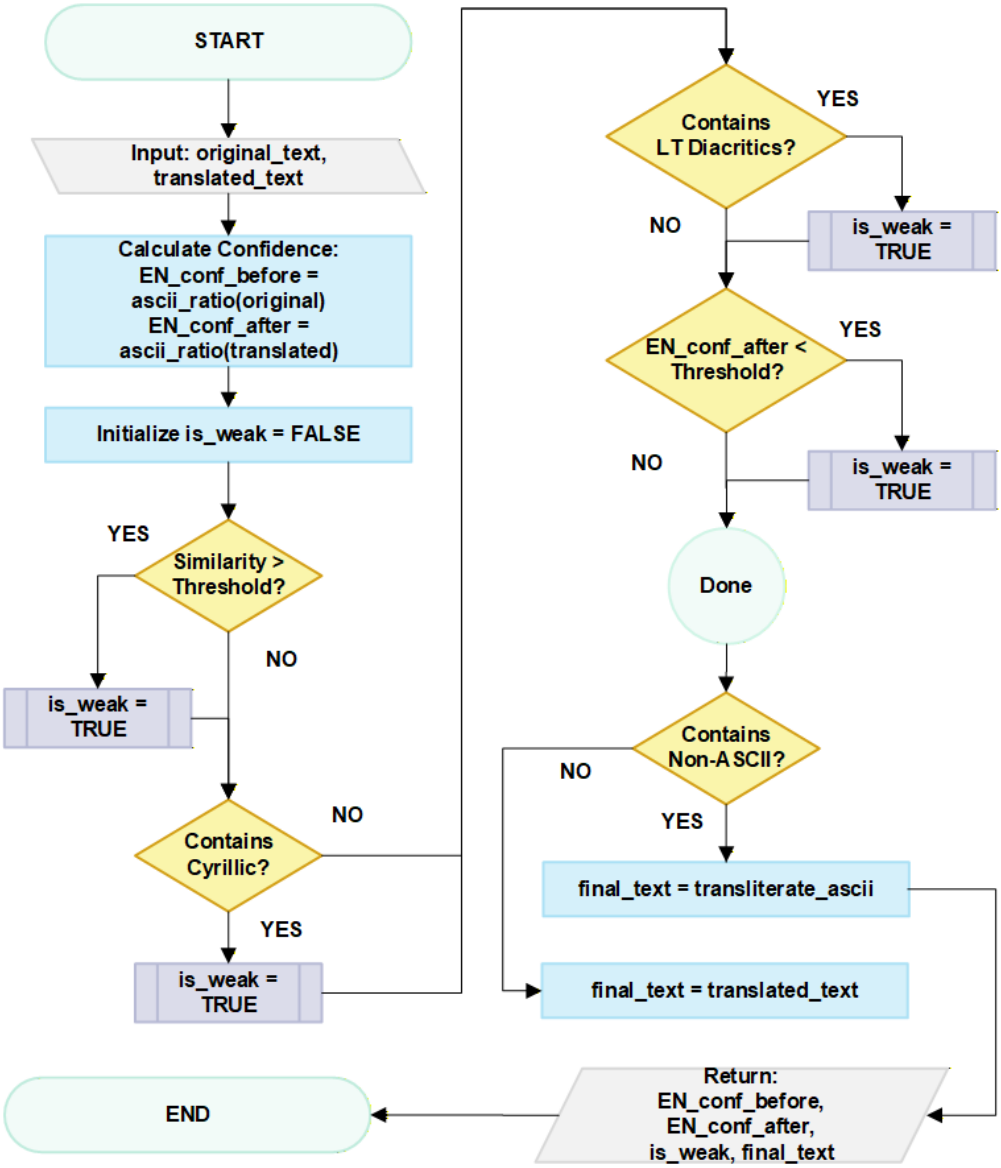


Figure 8. Translation throughput comparison across baseline, quantized, and optimized pipeline configurations.

The reduction in total execution time achieved by quantization and data flow optimization is illustrated in Figure 9. The translation throughput of Russian fragments in the

RU→EN stage increased to 25.00 fragments per second, while the translation throughput of Lithuanian fragments in the LT→EN stage reached 20.42 fragments per second. Accordingly, the RU→EN stage was completed in 28 min 17 s, the LT→EN stage in 13 min 22 s, and the entire two-stage translation process required only 42 min 07 s. The effective system throughput reached 33.35 incident records per second, exceeding the baseline experiment by more than a factor of twelve.

In terms of translation quality, the third experiment not only compensated for the decrease observed in the second stage but also surpassed the results of the baseline configuration. The impact of different experimental configurations on BLEU scores is illustrated in Figure 10. The variation of COMET scores across the three experimental stages is presented in Figure 11. The BLEU score increased to 42.73 and the COMET score to 82.97, corresponding to improvements of (+15.71%) and (+4.43%), respectively, compared to the first experiment. Since all results were obtained using an identical evaluation methodology and the same dataset, this improvement in translation quality is attributed directly to the applied data flow optimization strategies.

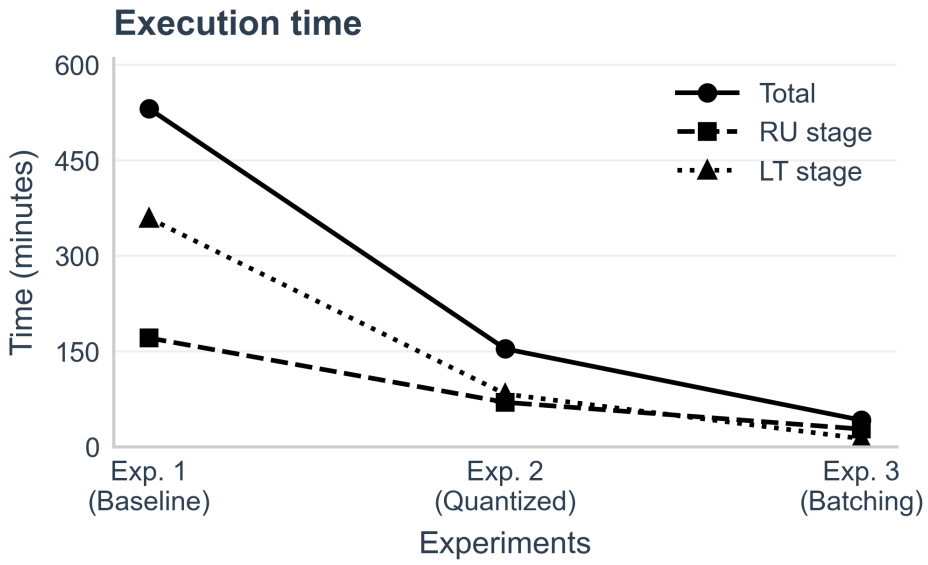


Figure 9. Execution time comparison of RU→EN and LT→EN stages across experimental configurations.

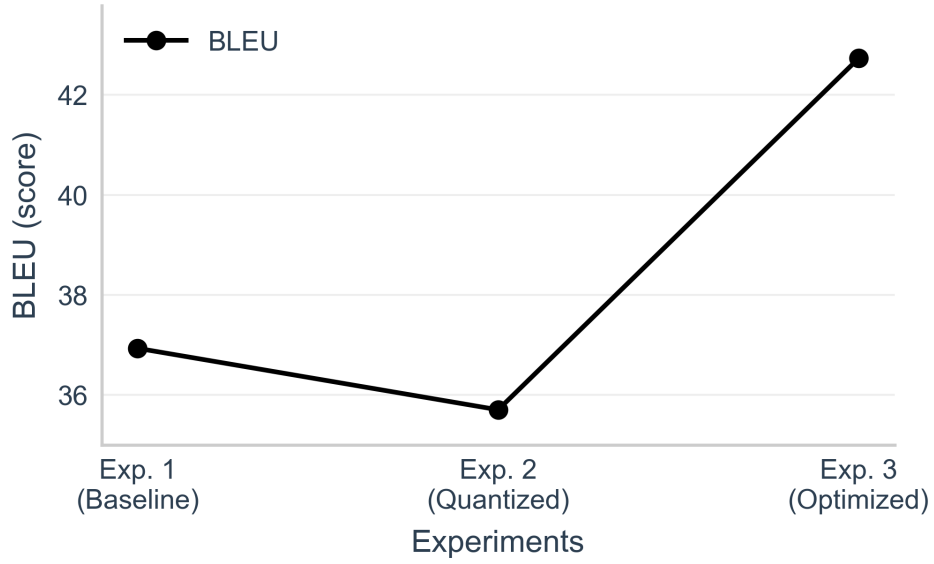


Figure 10. BLEU score comparison across baseline, quantized, and data-flow-optimized experiments.

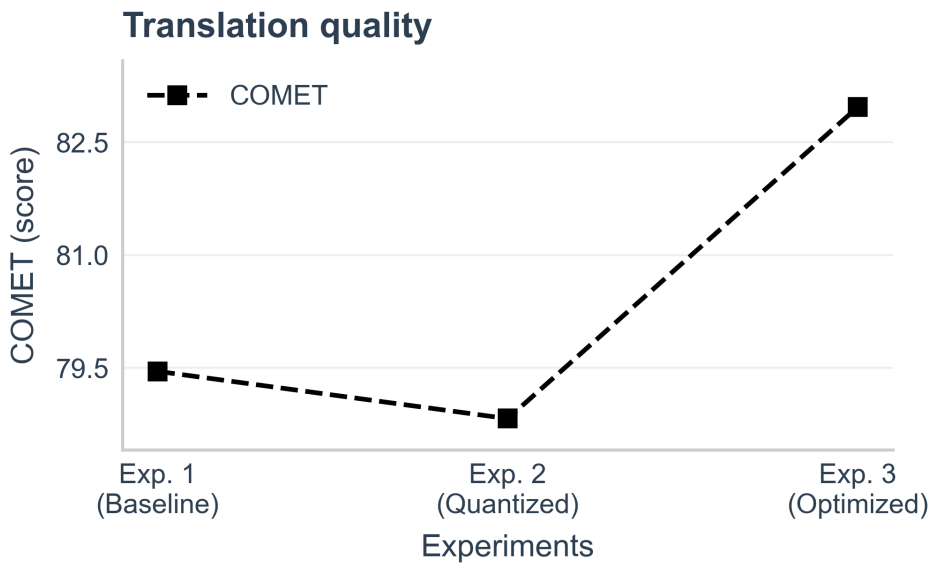


Figure 11. COMET score comparison across baseline, quantized, and data-flow-optimized experiments.

6. Discussion

The results confirm that a staged cascaded NMT strategy is theoretically relevant for code-switched operational texts where language identification is unreliable. Unlike general-domain multilingual corpora, ITSM incident descriptions contain short, noisy, and structurally irregular fragments that combine Lithuanian, Russian, English, technical identifiers, transliterated words, and incomplete syntax. Under these conditions, the RU→EN and LT→EN cascade provides a deterministic way to decompose multilingual input into sequential processing stages and reduce dependence on unstable language identification.

From a practical perspective, the proposed pipeline is relevant for enterprise ITSM environments because it supports local translation without external APIs, cloud-based inference, or network-dependent services. This is important for organizations where incident descriptions may contain internal system identifiers, infrastructure details, user-related information, and failure patterns. Normalized English-language output can support downstream ITSM tasks such as incident classification, duplicate detection, SLA-oriented reporting, root-cause analysis, and retrospective service improvement.

Compared with previous studies on multilingual NMT and code-switched machine translation, this study does not introduce a new neural translation architecture. Specifically, prior studies such as Chen et al. [4], Khanuja et al. [15], Khatri et al. [8], Nguyen et al. [11], and Srivastava and Singh [14] have mainly addressed benchmark-based evaluation, general-domain code-switching, social media text, or multilingual model behavior on mixed-language inputs. Other studies, including Fan et al. [6], Dabre et al. [2], Dabre et al. [19], and Tiedemann et al. [10], have focused on multilingual NMT architectures, multilingual transfer, large-scale multilingual models, open-source MT ecosystems, and broader model-level translation capabilities. In contrast, the contribution of the present study lies in the system-level integration and empirical evaluation of existing open-source NMT models under operational constraints. Specifically, this study evaluates real-world LT/RU/EN ITSM incident data under CPU-only, air-gapped, deterministic, and reproducibility-oriented conditions.

Several limitations should be acknowledged. The original ITSM dataset cannot be made publicly available because it contains operationally sensitive and organization-specific information. The evaluation is based on one organizational environment and one specific language combination, which limits opportunities for generalization. The

expert reference translation provides a practical evaluation benchmark but remains constrained by manual validation effort. In addition, the Man-hours field should be interpreted only as an administrative proxy for work effort and was not used as a primary translation quality metric.

7. Conclusions

This study evaluated a deterministic two-stage RU→EN and LT→EN cascaded neural machine translation pipeline for normalizing multilingual ITSM incident text under air-gapped and CPU-only execution constraints. The results show that the proposed approach is suitable for real-world incident descriptions characterized by code-switching, fragmentary syntax, heterogeneous terminology, and unreliable language identification.

The main finding is that translation performance and operational efficiency depend not only on model-level optimization but also on the structure of the complete data processing workflow. Normalization, segmentation, deduplication, adaptive batching, quantization, and language-aware routing jointly reduce excessive inference operations and improve the practical applicability of the pipeline without demanding changes to the underlying NMT models.

The main contribution of this study is therefore not a new neural translation architecture, but a reproducible system-level implementation and empirical evaluation of an air-gapped cascaded NMT workflow for privacy-constrained ITSM environments. Future research should evaluate the proposed pipeline in three specific directions. First, the translated and normalized incident texts should be tested in downstream ITSM tasks such as incident classification, duplicate incident detection, SLA-risk prediction, and causal relationship analysis. Second, the approach should be validated using datasets from additional organizations and other language combinations to assess whether the observed efficiency and quality gains generalize beyond the LT/RU/EN setting. Extending the approach to other languages would require language-specific adaptation of the translation stages, routing logic, segmentation rules, and evaluation procedure. For example, an extension to Chinese-language incident descriptions could be implemented by introducing an additional or alternative ZH→EN translation stage using a local, open-source model such as `Helsinki-NLP/opus-mt-zh-en`. However, such an extension would also require replacing the current LT/RU-oriented routing logic with Han-character-aware detection rules, including the detection of Chinese characters, Chinese punctuation, and mixed Latin-Chinese technical expressions. The segmentation module would also need to be adapted because Chinese text does not rely on whitespace-delimited word boundaries in the same way as Lithuanian, Russian, or English. Therefore, sentence and fragment segmentation would need to account for Chinese punctuation, character-level boundaries, and embedded IT terms written in Latin script. Finally, a new expert-validated reference sample would be required for the Chinese setting, and BLEU and COMET evaluations should be repeated against corresponding ZH→EN reference translations. These adaptations indicate that the proposed architecture is transferable at the system-design level, but not language-independent without script-aware routing, language-specific segmentation, and renewed validation. Third, future work should compare alternative local NMT models, routing strategies, and segmentation rules under the same CPU-only and air-gapped constraints, using BLEU, COMET, execution time, throughput, and downstream task performance as evaluation criteria.

Supplementary Materials: The following supporting information can be downloaded at https://www.mdpi.com/article/10.3390/make8070191/s1. The package contains a notebook-based source-code package of the proposed air-gapped cascaded NMT pipeline. It includes notebooks covering data preprocessing, field identification, text normalization, segmentation, deduplication, baseline

CPU-only translation, dynamic quantization, RAM-adaptive batching, language-aware routing, cascaded RU→EN and LT→EN translation logic, BLEU/COMET evaluation, and aggregate result visualization. The package also includes aggregate figure files used to report the experimental results. The original operational ITSM dataset, translated incident texts, expert reference translations, translation caches, internal infrastructure paths, credentials, organization-specific identifiers, and production deployment details are not included due to confidentiality, data protection, and the organization's security restrictions.

Author Contributions: Conceptualization, R.J. and D.M.; methodology, R.J.; software, R.J.; validation, R.J. and D.M.; formal analysis, R.J.; investigation, R.J.; resources, R.J.; data curation, R.J.; writing—original draft preparation, R.J.; writing—review and editing, R.J. and D.M.; visualization, R.J.; supervision, D.M.; project administration, R.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset analyzed in this study contains operational IT service management incident records and is not publicly available due to confidentiality, data protection, and the organization's security restrictions. Aggregated experimental results supporting the conclusions of this article are included within the article. A notebook-based source-code package is provided as Supplementary Material under the file name "make-4366232-supplementary.zip". Further inquiries may be directed to the corresponding author, subject to applicable confidentiality and institutional restrictions.

Acknowledgments: The authors would like to express sincere gratitude to Mindaugas Bereiša for insightful discussions and valuable feedback that contributed to the refinement of the research scope and methodological perspective of this work. The authors greatly appreciate his intellectual input and support during the early stages of this research.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Hidayatullah, A.F.; Qazi, A.; Lai, D.T.C.; Apong, R.A. A systematic review on language identification of code-mixed text: Techniques, data availability, challenges, and framework development. *IEEE Access* **2021**, *9*, 134162–134186. [CrossRef]
2. Dabre, R.; Chu, C.; Kunchukuttan, A. A survey of multilingual neural machine translation. *ACM Comput. Surv.* **2020**, *53*, 99. [CrossRef]
3. Cohn-Gordon, R.; Goodman, N.D. Lost in machine translation: A method to reduce meaning loss. In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL), Minneapolis, MN, USA, 2–7 June 2019; pp. 437–447.
4. Chen, S.; Aguilar, G.; Srinivasan, A.; Diab, M.; Solorio, T. CALCS 2021 shared task: Machine translation for code-switched data. In Proceedings of the 5th Workshop on Computational Approaches to Linguistic Code-Switching (CALCS), Online, 11 June 2021; pp. 1–12.
5. Sheth, R.; Sinha, S.R.; Patil, M.; Beniwal, H.; Singh, M. Beyond monolingual assumptions: A survey of code-switched NLP in the era of large language models. *arXiv* **2023**, arXiv:2310.01891.
6. Fan, A.; Bhosale, S.; Schwenk, H.; Ma, Z.; El-Kishky, A.; Goyal, S.; Baines, M.; Celebi, O.; Wenzek, G.; Chaudhary, V.; et al. Beyond English-centric multilingual machine translation. *J. Mach. Learn. Res.* **2021**, *22*, 1–48.
7. Kudo, T. Subword regularization: Improving neural network translation models with multiple subword candidates. In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL), Melbourne, Australia, 15–20 July 2018; pp. 66–75.
8. Khatri, J.; Srivastava, V.; Vig, L. Can you translate for me? Code-switched machine translation with large language models. *arXiv* **2024**, arXiv:2401.04661.
9. Tiedemann, J.; Thottingal, S. OPUS-MT—Building open translation services for the world. In Proceedings of the 22nd Conference of the European Association for Machine Translation (EAMT), Lisboa, Portugal, 3–5 November 2020; pp. 479–480.
10. Tiedemann, J.; Aulamo, M.; Bakshandaeva, D.; Boggia, M.; Grönroos, S.A.; Nieminen, T.; Raganato, A.; Scherrer, Y.; Vázquez, R.; Virpioja, S. Democratizing neural machine translation with OPUS-MT. *Lang. Resour. Eval.* **2024**, *58*, 713–755.

11. Nguyen, L.; Mayeux, O.; Yuan, Z. Code-switching input for machine translation: A case study of Vietnamese–English data. *Int. J. Multiling.* **2024**, *21*, 2268–2289.
12. Zhang, W.; Dai, L.; Liu, J.; Wang, S. Improving many-to-many neural machine translation via selective and aligned online data augmentation. *Appl. Sci.* **2023**, *13*, 3946. [CrossRef]
13. Licht, H.; Sczepanski, R.; Laurer, M.; Bekmuratovna, A. *No More Cost in Translation: Validating Open-Source Machine Translation for Quantitative Text Analysis*; University of Bonn and University of Cologne, Reinhard Selten Institute (RSI): Bonn, Germany, 2024.
14. Srivastava, V.; Singh, M. PHINC: A parallel Hinglish social media code-mixed corpus for machine translation. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL), Florence, Italy, 28 July–2 August 2019; pp. 3742–3751.
15. Khanuja, S.; Dandapat, S.; Srinivasan, A.; Sitaram, S.; Choudhury, M. GLUECoS: An evaluation benchmark for code-switched NLP. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL), Online, 5–10 July 2020; pp. 3575–3585.
16. Auer, P. Language contact and language change: Pragmatic factors. In *The Handbook of Applied Linguistics*; Blackwell: Oxford, UK, 2004.
17. Myers-Scotton, C. *Multiple Voices: An Introduction to Bilingualism*; Blackwell: Malden, MA, USA, 2006.
18. Masoud, M.; Torregrosa, D.; Buitelaar, P.; Arčan, M. Back-translation approach for code-switching machine translation: A case study. In Proceedings of the 1st Workshop on Technologies for Machine Translation of Low-Resourced Languages (LoResMT), Boston, MA, USA, 21 March 2018; pp. 47–54.
19. Dabre, R.; Chu, C.; Kunchukuttan, A. A comprehensive survey of multilingual neural machine translation. *arXiv* **2020**, arXiv:2001.01115. [CrossRef]
20. Zoph, B.; Yuret, D.; May, J.; Knight, K. Transfer learning for low-resource neural machine translation. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Austin, TX, USA, 1–5 November 2016; pp. 1568–1575.
21. Gu, J.; Wang, Y.; Chen, Y.; Cho, K.; Li, V.O.K. Meta-learning for low-resource neural machine translation. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Brussels, Belgium, 31 October–4 November 2018; pp. 3622–3631.
22. Bapna, A.; Chen, M.X.; Firat, O.; Cao, Y.; Wu, Y. Training deeper neural machine translation models with transparent attention. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Brussels, Belgium, 31 October–4 November 2018; pp. 3028–3033.
23. Sennrich, R.; Haddow, B.; Birch, A. Improving neural machine translation models with monolingual data. In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL), Berlin, Germany, 7–12 August 2016; pp. 86–96.
24. Fadaee, M.; Bisazza, A.; Monz, C. Data augmentation for low-resource neural machine translation. In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL), Vancouver, BC, Canada, 30 July–4 August 2017; pp. 567–573.
25. Warstadt, A.; Singh, A.; Bowman, S.R. Neural network acceptability judgments. *Trans. Assoc. Comput. Linguist.* **2019**, *7*, 625–641. [CrossRef]
26. Johnson, M.; Schuster, M.; Le, Q.V.; Krikun, M.; Wu, Y.; Chen, Z.; Thorat, N.; Viégas, F.; Wattenberg, M.; Corrado, G.; et al. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Trans. Assoc. Comput. Linguist.* **2017**, *5*, 339–351.
27. Yan, M.; Zhang, H.; Jin, D.; Zhou, J.T. Multi-source meta transfer for low resource multiple-choice question answering. In Proceedings of the AAAI Conference on Artificial Intelligence, Online, 2–9 February 2021; pp. 14216–14224.
28. Abadi, M.; Chu, A.; Goodfellow, I.; McMahan, H.B.; Mironov, I.; Talwar, K.; Zhang, L. Deep learning with differential privacy. In Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS), Vienna, Austria, 24–28 October 2016; pp. 308–318.
29. Zhang, H.K.; Sasano, R.; Takeda, K.; Wong, Z.S.-Y. Development of a medical incident report corpus with intention and factuality annotation. In Proceedings of the International Conference on Language Resources and Evaluation (LREC), Marseille, France, 11–16 May 2020; pp. 4578–4584.
30. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.-J. BLEU: A method for automatic evaluation of machine translation. In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, PA, USA, 6–12 July 2002; pp. 311–318.
31. Hilbig, I.; Jakaitė-Bulbukienė, K.; Daraškienė, I.; Žurauskaitė, E. (Eds.) *Terp Taikomosios Kalbotėros Barų: Mokslinių Straipsnių Rinkinys Profesorės Meilutės Ramonienės Jubiliejui*; Vilniaus Universiteto Leidykla: Vilnius, Lithuania, 2023.