Doctoral Dissertation

# A Multi-Aspect Reinforcement Learning Approach to Improving Entity-Aware Machine Translation

TRIEU Hoang An

Supervisor : NGUYEN Le Minh

Division of Advanced Institute of Science and Technology
Japan Advanced Institute of Science and Technology
Information Science
June, 2026

**Abstract**

Language barriers have long hindered effective knowledge sharing and the advancement of society. In recent years, the rapid development of machine translation, one of the most complex tasks in natural language processing, has contributed significantly to overcoming this obstacle. The impressive performance of modern translation systems demonstrates their capability in handling general-domain text with high accuracy. However, when it comes to terminology-sensitive domains such as medicine or law, these systems often fail to accurately translate domain-specific terms. This limitation poses a serious issue: even a single mistranslated term can alter the meaning of the original text and lead to severe misinterpretations, especially in high-stakes contexts. Motivated by this challenge, the goal of this thesis is to develop a machine translation system capable of reliably and correctly translating terminologies in specialized domains.

The first challenge lies in ensuring that the system maintains strong general-domain translation performance after domain adaptation. Fine-tuning a model for a specific field may cause it to lose previously learned knowledge, resulting in degraded performance on general text. To address this, we propose a multiple-aspect reinforcement learning approach that evaluates and optimizes the model using multiple complementary criteria. This technique allows us to control knowledge retention inside the model, preventing catastrophic forgetting while simultaneously enhancing its ability to correctly translate specialized terminology. We further introduce clear definitions of the proposed aspects and describe how they can be incorporated into the training process to improve overall system performance.

The second challenge pertains to the limited availability of high-quality data for assessing and improving terminology translation accuracy. Creating datasets that precisely identify and evaluate domain-specific terminology usage is costly and resource-intensive. To overcome this, we propose a data generation framework for producing a high-quality silver-standard dataset to support training and fine-tuning. Additionally, we outline criteria for constructing a smaller but reliable gold test dataset used for evaluation and testing.

In summary, this thesis aims to enhance machine translation systems by improving their ability to translate specialized terminology while preserving strong performance in general-domain translation. Through a combination of reinforcement learning and efficient data generation methods, our approach provides a promising direction for advancing terminology-aware machine translation.

**Keywords:** *Machine Translation, , Large Language Models, , Entity-Aware Machine Translation, , Reinforcement Learning, , Dataset Construction.*

# Acknowledgments

First and foremost, I am deeply grateful to my supervisor, Professor Nguyen Le Minh at the Japan Advanced Institute of Science and Technology (JAIST), for the opportunity to pursue this work and for shaping my understanding of research. His guidance helped me identify a topic that fit my interests and strengths, and his patience and encouragement sustained me through setbacks.

I thank Professor Kiyoaki Shirai, Professor ITTOO Ashwin, Professor Hasegawa Shinobu, Associate Professor Masnizah Moh, Associate Professor Mizumoto Masaharu, and Associate Professor Teeradaj Racharak for their constructive questions and suggestions, which significantly improved this research.

I am indebted to Dr. Tran Duc Vu and Dr. Nguyen Minh Phuong for their candid advice and generous support, which helped me both in life and in my research.

I also thank the JAIST staff and the Football Clubs for fostering a supportive environment for work, study, and well-being.

My sincere appreciation goes to all members of the Nguyen Laboratory for their daily help and insightful discussions during my time at JAIST.

Finally, I express my deepest love and gratitude to my family, especially my grandmother, for their unwavering understanding and support. They are my source of strength; without their unconditional support, I could not have completed this thesis.

# Contents

# List of Figures

# List of Tables

# Notation

| Symbol | Description |
|---|---|
| *Machine Translation* | |
| $x$ | Source sentence |
| $y^\star$ | Reference (gold) translation |
| $\hat{y}$ | Model hypothesis (generated translation) |
| $\mathcal{D} = \{(x_i, y_i^\star)\}$ | Training dataset of parallel sentence pairs |
| $K$ | Group size (number of candidate hypotheses sampled per source) |
| $M$ | Number of hypotheses in a GRPO group |
| *Reinforcement Learning* | |
| $(\mathcal{S}, \mathcal{A}, p, r, \gamma)$ | Markov decision process (state space, action space, transition kernel, reward function, discount factor) |
| $\pi_\theta(a \mid s)$ | Parameterized policy with parameters $\theta$ |
| $J(\theta)$ | Expected discounted return |
| $Q^\pi(s, a)$ | Action-value function under policy $\pi$ |
| $V^\pi(s)$ | State-value function under policy $\pi$ |
| $A^\pi(s, a)$ | Advantage function: $Q^\pi(s, a) - V^\pi(s)$ |
| $\hat{A}_t$ | Estimated advantage at time step $t$ |
| $\hat{A}_t^{\mathrm{GAE}(\gamma,\lambda)}$ | Generalized advantage estimate |
| $\delta_t$ | Temporal-difference error: $r_t + \gamma V_\phi(s_{t+1}) - V_\phi(s_t)$ |
| $\gamma$ | Discount factor in MDP |
| $\lambda$ | Trace-decay parameter in GAE |
| $\rho_t(\theta)$ | Importance-sampling ratio: $\pi_\theta(a_t \mid s_t)/\pi_{\theta_{\mathrm{old}}}(a_t \mid s_t)$ |
| $\epsilon$ | Clipping parameter in PPO |
| $L^{\mathrm{CLIP}}(\theta)$ | PPO clipped surrogate objective |
| $D_{\mathrm{KL}}$ | Kullback–Leibler divergence |
| $\delta$ | KL radius (trust-region bound in TRPO) |
| $c_V$ | Value-function loss coefficient |
| $c_H$ | Entropy bonus coefficient |
| $\mathcal{H}(\pi)$ | Entropy of policy $\pi$ |
| $V_\phi(s)$ | Learned value-function approximation with parameters $\phi$ |
| *GRPO (Group Relative Policy Optimization)* | |
| $\tilde{A}_j$ | Centered rank-derived advantage for candidate $j$ |
| $\mathrm{rank}(j)$ | Rank of candidate $j$ within the group |
| $R_j$ | Evaluator/reward score for candidate $j$ |
| $\tau$ | Temperature parameter controlling rank-advantage sharpness |
| $L^{\mathrm{GRPO}}(\theta)$ | GRPO clipped objective |
| *Reward Functions* | |
| $R_{\mathrm{total}}$ | Composite reward: $\alpha R_{\mathrm{BLEU}} + \beta R_{\text{M-ETA}} + \delta R_{\mathrm{attn}} + \eta R_{\mathrm{dep}}$ |
| $\alpha, \beta, \delta, \eta$ | Mixing coefficients for reward components |
| $R_{\mathrm{BLEU}}$ | Lexical matching reward (sentence-level BLEU / 100) |
| $R_{\text{M-ETA}}$ | Entity matching reward (recall of gold entities) |

*Continued on next page*

| Symbol | Description |
|---|---|
| $R_{\text{attn}}$ | Attention similarity reward |
| $R_{\text{dep}}$ | Dependency-graph syntactic similarity reward |
| $R_{\text{cfg}}$ | Constituency-graph (CFG) syntactic similarity reward |
| $R_{\text{syn}}$ | Syntactic similarity reward |
| $\lambda_{\text{ent}}, \lambda_{\text{attn}}, \lambda_{\text{syn}}$ | Per-aspect reward scaling weights |
| $r_j^{\text{ent}}, r_j^{\text{attn}}, r_j^{\text{syn}}$ | Scaled per-sample rewards for entity, attention, syntactic aspects |
| $a_j^{\text{ent}}, a_j^{\text{attn}}, a_j$ | Centered per-sample advantages for each reward aspect |
| *Entity Matching* | |
| $E_{\text{ref}}$ | Set of gold entities: $\{e_1, \ldots, e_n\}$ |
| $E_{\text{hyp}}$ | Set of entity mentions detected in hypothesis $\hat{y}$ |
| *Attention Similarity* | |
| $A^{(\ell,h)}$ | Attention matrix for layer $\ell$, head $h$ ($\in \mathbb{R}^{T_{\text{tgt}} \times T_{\text{src}}}$) |
| $\bar{A}$ | Aggregated, row-normalized attention matrix |
| $A^\star$ | Reference alignment from external aligner |
| $w_{\ell,h}$ | Nonnegative head/layer aggregation weights |
| $L, H$ | Number of layers, number of attention heads |
| $T_{\text{src}}, T_{\text{tgt}}$ | Length of source and target token sequences |
| $\langle \cdot, \cdot \rangle_F$ | Frobenius inner product |
| $\varepsilon$ | Small constant for numerical stability |
| *Syntactic Graphs* | |
| $G = (V, E, \ell)$ | Labeled directed graph (nodes, edges, label function) |
| $G_{\text{dep}}(y)$ | Dependency graph of sentence $y$ |
| $G_{\text{cfg}}(y)$ | Constituency (CFG) graph of sentence $y$ |
| $G^{(k)}$ | $k$-hop ego subgraph around an entity span |
| $\mathcal{E}$ | Entity span in a sentence |
| $\mathcal{N}_G^{(k)}(e)$ | $k$-hop neighborhood of entity $e$ in graph $G$ |
| $K_{\text{WL}}$ | Weisfeiler–Lehman subtree kernel |
| $\widetilde{K}_{\text{WL}}$ | Normalized WL kernel similarity ($\in [0, 1]$) |
| $K_{\text{CFG}}$ | Tree kernel for constituency trees |
| $k$ | Number of WL refinement rounds / hop distance |
| *Graph Distance Metrics* | |
| $\text{GED}(G_1, G_2)$ | Graph edit distance |
| $\lambda(L_G)$ | Eigenvalues of the (normalized) Laplacian of graph $G$ |
| *Transformer Architecture* | |
| $X$ | Input embeddings |
| $T$ | Length of the input sequence |
| $d_{\text{emb}}$ | Dimensionality of embedding vectors |
| $d$ | Hidden representation size |
| $Q^h, K^h, V^h$ | Query, Key, and Value matrices for attention head $h$ |
| $W_Q^h, W_K^h, W_V^h$ | Learnable weight matrices for Q/K/V projections |
| $C^h$ | Context vector for attention head $h$ |
| $P_{i,2j}, P_{i,2j+1}$ | Sinusoidal positional encoding components |
| $e_{t,i}$ | Alignment score between decoder state $t$ and encoder state $i$ |
| $h_d^t$ | Decoder hidden state at time step $t$ |

*Continued on next page*

| Symbol | Description |
|---|---|
| $h_e^i$ | Encoder hidden state at position $i$ |
| $W_a, W_1, W_2$ | Learnable weight matrices in attention mechanisms |

# Chapter 1

# Introduction

## 1.1 Background and Motivation

Machine translation (MT) has long been regarded as one of the most challenging and multifaceted tasks in natural language processing (NLP). Over several decades, the field has evolved from rule-based and statistical approaches to the current generation of neural architectures, culminating in the development of the Transformer and the subsequent rise of large language models (LLMs). These advances have dramatically improved translation performance across a wide range of general-domain texts, enabling models to capture long-range dependencies, perform sophisticated contextual reasoning, and generate fluent outputs comparable to human translation in many everyday scenarios. Despite these notable improvements, the limitations of contemporary MT systems become readily apparent when they are applied to specialized or closed-domain settings—particularly in domains such as medicine, biomedical research, law, or engineering—where the precise and unambiguous translation of named entities and technical terminology is paramount. In such contexts, even a seemingly small discrepancy in the translation of a domain-specific entity can propagate through the text, altering the intended meaning, hindering information extraction, or in sensitive domains like healthcare, potentially leading to harmful misunderstandings. In contrast to general vocabulary, which is often flexible and may allow for multiple valid context-dependent renderings, domain-specific entities are typically rigid, semantically dense, and highly sensitive to mistranslation. This rigidity underscores their importance and highlights the unique challenges they pose for MT systems, which must simultaneously preserve fidelity to critical terminology while maintaining overall linguistic fluency.

These challenges have motivated the growth of entity-aware machine translation (EAMT), a specialized subfield dedicated to enhancing how MT systems identify, represent, and accurately translate named entities and domain-specific terminology. Over the years, researchers have proposed a wide range of strategies to address entity-related errors, each grounded in different assumptions about how models handle lexical specificity. One class of approaches integrates external knowledge sources—such as structured ontologies, bilingual dictionaries, terminology databases, or curated glossaries—directly into the translation pipeline. These methods typically annotate source sentences with explicit entity information or supply reference translations for key terms, allowing the model to anchor its predictions to authoritative lexical resources. While effective in scenarios where terminology is stable and well-documented, such methods depend heavily on the availability and quality of these external resources; coverage

gaps or inconsistencies in terminology lists often lead to brittle behavior or failure to generalize to unseen entities.

A second line of work focuses on modifying the internal architecture of neural translation models themselves. Techniques in this category inject constraints into the attention mechanism to force the model to focus more strongly on entity spans, introduce entity embeddings or feature vectors that encode entity type and boundary information, or design decoder-side constraints that ensure strict adherence to predefined entity translations. These methods are attractive because they integrate entity awareness directly into the model's learned representations, enabling fine-grained control over how the model processes specialized terms. However, such architectural interventions can inadvertently narrow the model's flexibility: enforcing strong constraints during decoding can interfere with the natural flow of language generation, sometimes degrading grammaticality, fluency, or contextual appropriateness—especially in general-domain settings where rigid entity handling is unnecessary.

A third family of approaches relies on post-processing and post-editing mechanisms. Instead of altering the generation process, these methods adjust the output after translation by detecting and correcting inconsistent or incorrect entity renderings. These corrections may involve replacing mistranslated terms with dictionary-approved equivalents, aligning entity spans across languages, or enforcing consistency across multiple occurrences of an entity within a document. While post-editing avoids the complexities of modifying the underlying MT model, it operates only after errors have occurred, making it inherently reactive. As a result, it may struggle with subtle contextual nuances, disambiguation challenges, or cases where incorrect entity translations propagate deeper semantic errors throughout the sentence.

Although each of these approaches has demonstrated meaningful gains in controlling entity translation, they also introduce important limitations. Systems that depend on fixed terminology glossaries may be unable to adapt when domain boundaries shift, when new terminology emerges, or when entities exhibit variation across contexts. Architecturally constrained models risk overfitting to their entity-handling mechanisms, reducing their fluency or complicating general-domain translation tasks. Post-editing methods, while convenient, often lack the ability to correct deeper structural errors or ensure that entity translations are contextually justified. Collectively, these constraints reveal a common challenge: most existing EAMT methods achieve improvements in entity accuracy only by sacrificing some combination of fluency, general-domain robustness, or adaptability.

Therefore, there is a clear need for a more flexible and balanced translation system—one that can dynamically adapt to terminology-sensitive domains without compromising its overall translation quality in general-domain contexts. Such a system should be capable of learning entity-aware behavior that is both contextually grounded and linguistically fluent, preserving the strengths of modern neural architectures while addressing their weaknesses in handling critical domain-specific terminology.

## 1.2 Problem statement and Objectives

Despite the breadth of prior work on entity-aware machine translation, two fundamental challenges remain largely unresolved and form the primary focus of this thesis.

**Challenge 1: Data Scarcity and Annotation Burden** Entity-aware translation requires supervision that goes beyond ordinary sentence-level parallel data. In addition to aligned source and target sentences, the model often needs span-level annotations that mark named entities, their types, and their standardized equivalents or normalization codes, for example in medical terminology mapped to ICD or INN standards. Such annotations are costly to produce because they usually require domain expertise and careful validation. As a result, high-quality entity-annotated resources remain scarce in many specialized domains, including medicine, law, and technical translation. This data limitation makes it difficult for entity-aware systems to learn reliable entity-handling behavior and to generalize to terms that were not seen during training.

**Challenge 2: Catastrophic Forgetting During Domain Adaptation** When a pre-trained general-domain MT model is adapted to a specialized domain, improvements in entity translation often come at the cost of reduced performance on general text. This degradation, commonly referred to as *catastrophic forgetting*, occurs because fine-tuning or reinforcement learning places stronger emphasis on domain-specific signals and may gradually overwrite the broader linguistic knowledge acquired during pre-training. The model is therefore forced to balance two competing goals: becoming more accurate in the target domain while preserving its ability to translate general-domain text fluently and reliably. This balance is especially important in practical systems that must serve both specialized and general translation scenarios.

**Research Objective** We seek to address these two intertwined challenges by developing a method that improves the translation of named entities within specialized domains while simultaneously preserving, and ideally enhancing, the performance of MT systems on general-domain text. Rather than treating entity fidelity and general fluency as competing objectives, we integrate them through a reinforcement learning (RL) formulation with multi-objective optimization. In this formulation, the translation model is treated as the agent, and each generated translation hypothesis is treated as an output state to be evaluated. The state is judged from multiple aspects, including entity correctness, lexical fluency, semantic consistency, and structural adequacy, and the resulting multi-aspect reward is used to update the agent policy. Rather than enforcing rigid constraints during decoding or relying solely on symbolic guidance, our approach allows the model to learn nuanced translation behavior through interaction with a carefully designed reward structure. This framework encourages the system to treat entity translation not as an isolated task but as a component of a broader, context-aware decision-making process. By optimizing the model using a blend of entity-focused and holistic linguistic rewards, we enable the MT system to develop a more stable and adaptive translation capability, one that handles critical domain-specific terminology with heightened precision while preserving the flexibility and fluency associated with modern MT architectures. Ultimately, our approach aims to create a more balanced, robust, and generalizable translation model capable of excelling in both closed-domain and general-domain scenarios.

## 1.3 Contributions

Our main contributions are summarized as follows:

- **Reinforcement-learning-based framework for balanced optimization.** We propose a novel reinforcement learning (RL) framework that integrates multiple reward signals to jointly optimize named entity translation accuracy and overall translation fluency. Unlike prior methods that emphasize either entity fidelity or global quality, our framework explicitly balances these competing objectives. By combining sentence-level fluency rewards with entity-aware accuracy measures, the model is encouraged to produce translations that remain terminologically reliable while maintaining naturalness in both specialized and general-domain settings.

- **Formalization and analysis of entity-aware reward criteria.** We define and formalize a comprehensive set of reward criteria tailored to entity-aware machine translation. These criteria capture multiple dimensions of translation quality, including n-gram adequacy, semantic coverage, and fine-grained entity correctness. Through systematic ablation and interaction analyses, we investigate the individual and combined effects of each reward component, providing insights into the trade-offs between strict entity fidelity and general-language fluency. Our findings offer valuable guidance for designing future multi-objective optimization strategies in MT.

- **Construction and release of an entity-annotated MT dataset.** We develop and publicly release a supporting dataset specifically designed for training and evaluating entity-aware translation systems. The dataset includes aligned source–target pairs enriched with manually verified named entity annotations, enabling rigorous assessment of terminological accuracy and end-to-end MT performance. By making this resource available to the research community, we aim to facilitate reproducible experimentation and accelerate progress in terminology-sensitive machine translation.

## 1.4 Thesis structure

The remainder of this thesis is organized as follows:

- **Chapter 2** provides a comprehensive review of related work, including foundational research in machine translation, reinforcement learning techniques, prior MT systems enhanced through RL, and the evaluation metrics—particularly graph-based distance measures—that inform the algorithms developed later in this thesis.

- **Chapter 3** presents the data construction pipeline, detailing the framework used to generate the silver-standard dataset and the annotation criteria employed to create high-quality golden labels for the test set.

- **Chapter 4** introduces the multi-task entity-aware machine translation model proposed to address the challenges of this study. This section also reports the experimental results and offers an in-depth discussion of the model's performance, strengths, and limitations.

- **Chapter 5** describes the multi-aspect reinforcement learning method designed to fine-tune large language models for terminology-sensitive translation. This chapter elaborates on each reward aspect, the formulation of the reward functions, and the overall multi-objective optimization strategy.

- **Chapter 6** presents an ablation study isolating the impact of each component within the framework. The thesis concludes with key insights, broader implications for entity-aware MT, and recommendations for future research directions.

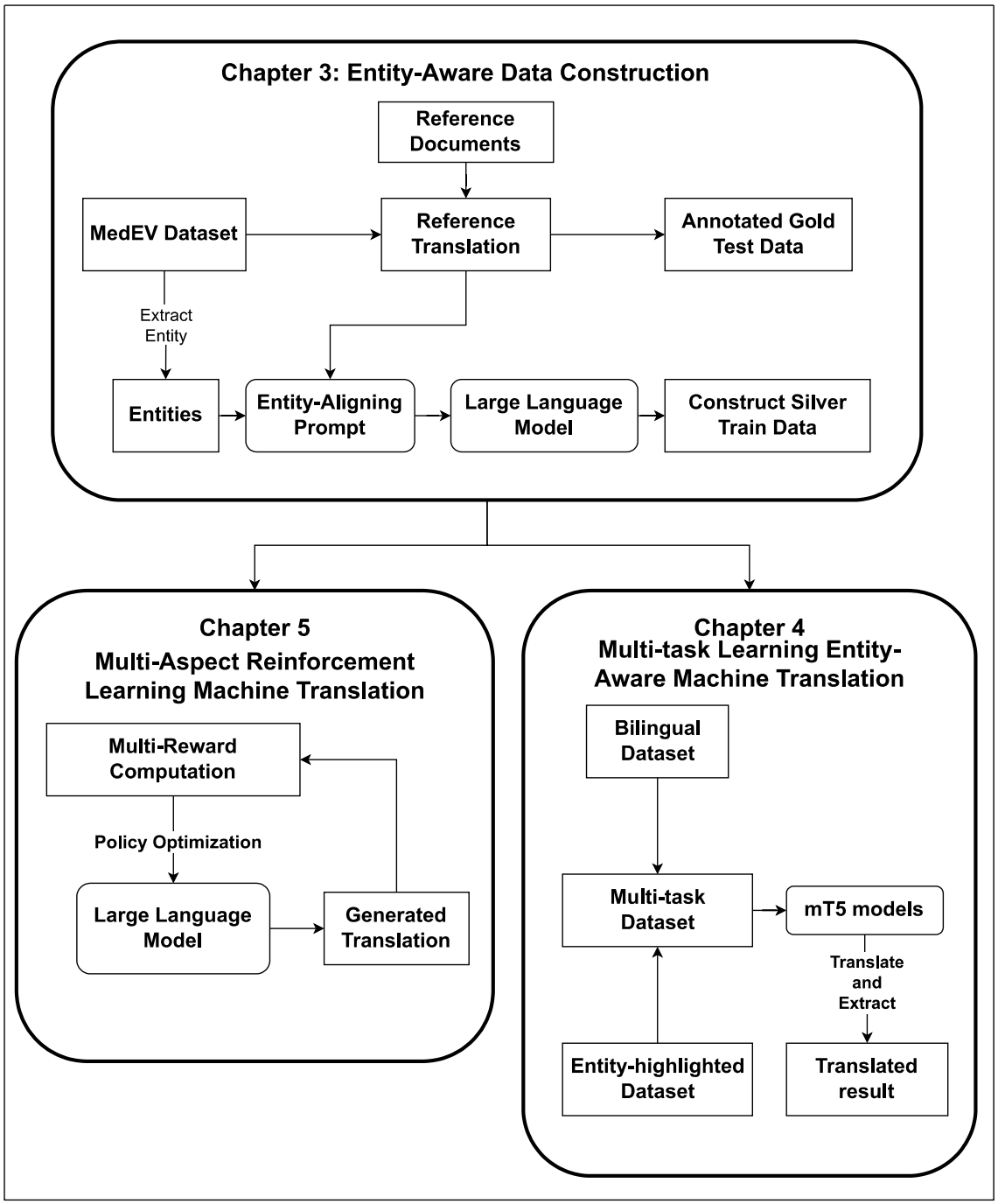


Figure 1.1: Dissertation Outline

# Chapter 2

# Preliminaries

The following chapter provides a comprehensive and in-depth examination of machine translation, reinforcement learning algorithms, and graph distance metrics. It begins by outlining the core concepts and theoretical foundations of each topic, followed by a discussion of their practical roles and applications within natural language processing.

## 2.1 Machine Translation

The idea of creating a system capable of conveying the same meaning across multiple languages dates back to the 17th century, when René Descartes first proposed the concept. Nevertheless, meaningful progress in Machine Translation (MT) did not emerge until the 1950s. In 1951, Yehoshua Bar-Hillel initiated one of the earliest formal research programs on MT at MIT, marking the beginning of systematic study in this field. Since then, numerous approaches, innovations, and technological advances have driven the rapid evolution of MT. Broadly, research developments from 1951 to the present can be grouped into three major system architectures: Rule-based Machine Translation, Statistical Machine Translation, and Neural Machine Translation.

### 2.1.1 Rule-Based Machine Translation

Rule-Based Machine Translation (RMT) was the earliest approach developed for automated translation, grounded in the idea that linguistic meaning can be systematically mapped between languages. In essence, RMT relies on predefined sets of correspondences: words or phrases in the source language are matched with equivalents in the target language. For example, the English term "dog" can be linked directly to the Japanese word "inu." Using these mappings, translation becomes a largely mechanical process of substituting source-language elements with their target-language counterparts. This method works reasonably well for straightforward sentences or highly controlled text.

However, the RMT approach faces significant limitations. Its reliance on rigid grammatical rules means that any translation is only as accurate as the rules themselves, which are notoriously difficult to construct and maintain. Languages contain numerous, sometimes conflicting syntactic rules, and resolving these conflicts can be extremely challenging. Moreover, RMT systems typically lack sensitivity to context, which is critical because the meaning of a word often depends on its surrounding text. Take the word "bank" as an example: in "He went to the bank to deposit money," it refers to a financial institution, whereas in "He walked along the river bank," it describes the land

beside a river. Rule-based systems cannot reliably differentiate these cases, leading to errors in translation.

Another major limitation arises from the inherent variability of natural language. Even within the constraints of formal grammar, human languages exhibit flexibility, idiomatic expressions, and nuances that are nearly impossible to capture exhaustively with fixed rules. Determining which rule takes precedence in ambiguous situations adds another layer of complexity. Consequently, while RMT laid the foundation for computational translation, its practical effectiveness is constrained by the challenges of context, ambiguity, and the sheer diversity of natural language usage.

### 2.1.2 Statistic Machine Translation

Statistical Machine Translation (SMT) emerged as a fundamentally different paradigm from traditional Rule-Based Machine Translation (RMT). Rather than relying on handcrafted linguistic rules, SMT builds translations from patterns learned directly from bilingual corpora. Its central assumption is that translation can be framed as a probabilistic decision-making process: given a source sentence, the system selects the target sentence with the highest likelihood according to statistical models derived from observed word and phrase correspondences. In practice, SMT replaces segments of the source text—whether individual words or multi-word expressions—with their statistically most probable counterparts in the target language. This data-driven foundation makes SMT more flexible, scalable, and easier to adapt to new domains than RMT, which requires substantial manual effort to expand rule sets. Furthermore, SMT systems can be improved through better data and refined statistical modeling rather than extensive linguistic engineering.

However, this reliance on corpus statistics also introduces notable weaknesses. SMT systems are highly sensitive to the quality of their training data; noisy or misaligned corpora can degrade translation accuracy significantly. Additionally, SMT struggles with long or syntactically complex sentences, where local phrase-based decisions fail to capture broader semantic dependencies. These limitations highlight two longstanding challenges for SMT: robustness to noisy datasets and the ability to handle long-range contextual information.

To address these shortcomings, a variety of enhancements and sub-components were developed, particularly within Phrase-Based Statistical Machine Translation (PBSMT). Techniques such as text normalization, sentence alignment, word alignment, phrase extraction, feature engineering, and n-gram language modeling became standard in SMT pipelines. PBSMT models translate by exploiting statistical relationships between phrases in the source and target languages, allowing them to produce more coherent and contextually appropriate translations than earlier word-for-word approaches. As a result, PBSMT achieved substantial improvements over earlier systems and became the dominant form of SMT for more than a decade.

Another key factor contributing to the rise of SMT was the introduction of word embedding techniques. Early SMT pipelines often relied on sparse, high-dimensional representations such as Term Frequency–Inverse Document Frequency (TF-IDF), which struggled to capture deeper semantic relationships between words. Word embeddings were developed to address this limitation by mapping each token to a dense numeric vector that encodes semantic and syntactic properties. Early embedding approaches, such as word2vec, GloVe, doc2vec, and fastText, constructed vector spaces in which similarity between vectors reflects meaningful relationships between words. These con-

tinuous representations allowed SMT systems to operate with richer contextual information, enabling more accurate alignment decisions and improved phrase selection. Before translation, SMT models embed tokens into these learned vector spaces, allowing subsequent statistical operations to exploit semantic proximity rather than relying solely on surface-level co-occurrence frequencies.

### 2.1.3 Neural Machine Translation

Despite the notable advances of Statistical Machine Translation (SMT), these systems still face significant limitations in capturing the full semantic context of input sentences, which can lead to translations that fail to convey the intended meaning. Moreover, SMT models are highly sensitive to noise in training data, and their reliance on statistical correlations makes them less robust when confronted with imperfect or sparse datasets. To overcome these challenges, researchers turned to deep learning architectures, giving rise to Neural Machine Translation (NMT). NMT leverages neural networks to learn complex representations of language, enabling more accurate and contextually aware translations. Over time, NMT models have evolved into two dominant architectural families: Convolutional Neural Network (CNN)-based NMT and Recurrent Neural Network (RNN)-based NMT.

#### Recurrent Neural Network-based NMT

Recurrent Neural Networks (RNNs) have been a cornerstone in many NLP tasks before their adoption in translation. Their sequential processing capability allows them to maintain the order of tokens and encode contextual information from entire sentences. This sequential modeling gives RNN-based NMT an advantage over earlier SMT systems, as it can better capture dependencies between words and generate more coherent translations.

RNNs compute a hidden state at each time step by combining the current input token with the previous hidden state. This iterative process ensures that the final hidden state encapsulates information about the entire input sequence while preserving the relative ordering of tokens. Building on this structure, Sutskever et al. [2] proposed a full encoder-decoder RNN model for translation, which significantly outperformed SMT baselines. The source sentence is first encoded into a fixed-length vector, which, along with a start-of-sentence token, guides the generation of the first target token. Subsequent tokens are generated recursively, with each step conditioned on the previous hidden state and the embedding of the last generated token.

However, vanilla RNNs struggle with long sequences due to the vanishing gradient problem, which causes information from the beginning of a sentence to be lost during encoding. To address this, Long Short-Term Memory networks (LSTMs) were introduced by Hochreiter et al. [3], incorporating gating mechanisms to selectively retain or discard information over long sequences. Each LSTM unit, or Gated Recurrent Unit (GRU), allows the model to focus on relevant contextual information, improving translation quality for longer sentences. Later advancements, such as the Fast-Forward RNN by Zhou et al. [4], extended LSTMs into deeper architectures capable of capturing more complex dependencies.

**Attention Mechanisms** A major breakthrough in RNN-based NMT came with the introduction of attention mechanisms. Attention allows the model to weigh the importance of different source tokens dynamically during decoding, enabling better handling of long and complex sentences. Bahdanau et al. [5] proposed the first atten-

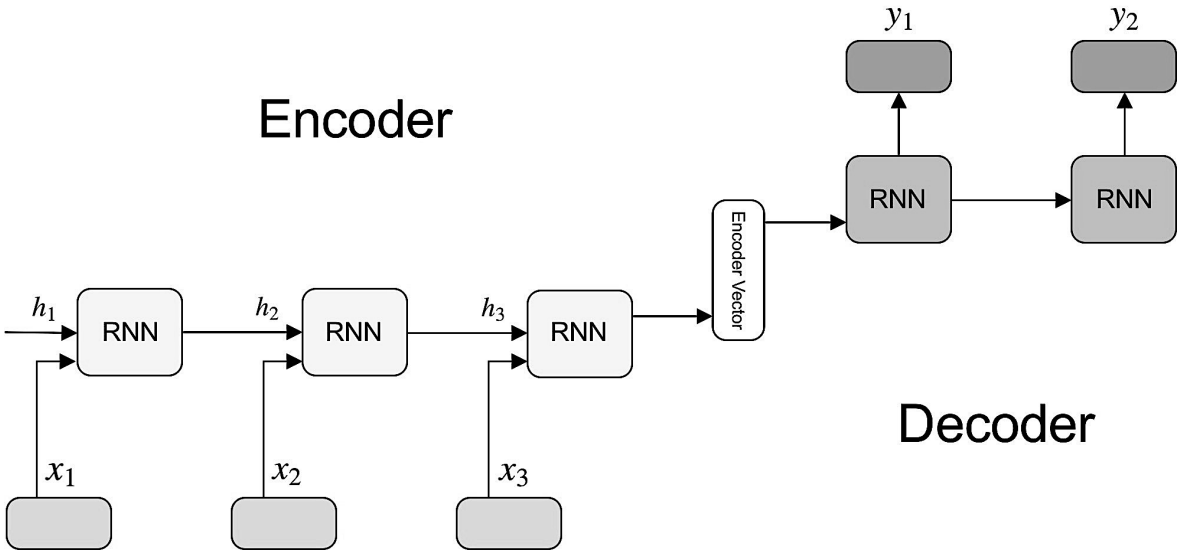


Figure 2.1: RNN structure

tion mechanism, computing a unique context vector at each decoding step based on the previous decoder hidden state and all encoder hidden states. This context vector guides the decoder in focusing on the most relevant portions of the input.

Luong et al. [6, 7] later refined this approach, offering multiple methods to compute the alignment scores between encoder and decoder states:

- **Dot product:** Computes the alignment by taking the dot product of the decoder and encoder hidden states.
$$e_{t,i} = h_d^{t-1} \cdot h_e^i \tag{2.1}$$
- **General:** Introduces a learnable weight matrix to scale the dot product between the states.
$$e_{t,i} = h_d^{t-1} \cdot W_a h_e^i \tag{2.2}$$
- **Concatenate:** Concatenates the decoder and encoder states, applies a nonlinear transformation, and then projects to a scalar alignment score.
$$e_{t,i} = W_1 \cdot \tanh(W_2([h_d^{t-1} : h_e^i])) \tag{2.3}$$

Unlike Bahdanau's approach, Luong's method combines encoder and decoder states before applying a shared weight matrix, simplifying computation while maintaining effective attention. The resulting context vector is then integrated into the decoding step to generate outputs that are more faithful to the source content while preserving fluency.

Through the combination of sequential modeling, gating mechanisms, and attention strategies, RNN-based NMT models achieved significant improvements over SMT, particularly in maintaining sentence-level coherence, capturing long-range dependencies, and translating context-sensitive expressions accurately.

**Convolutional Neural Network-based NMT**

Convolutional Neural Networks (CNNs) are widely recognized for their exceptional performance in computer vision tasks, such as object detection, image segmentation,

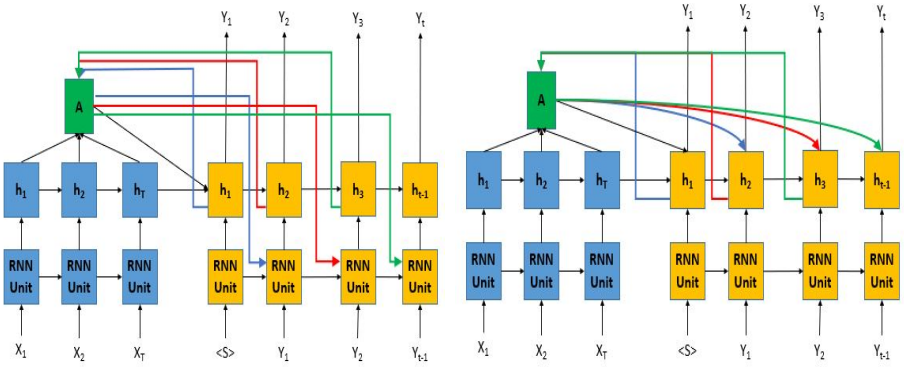


Figure 2.2: Demonstration of attention mechanisms proposed by Bahdanau et al. (left) and Luong et al. (right). <S> is the start-of-sentence token, h represents hidden state representations, and A represents attention weights computed from the hidden states.

and document layout analysis. Beyond vision, CNNs have also been explored in natural language processing, including machine translation. However, for a considerable period, CNN-based NMT models were generally outperformed by their RNN-based counterparts, primarily due to the sequential nature of language, which RNNs inherently model.

Early attempts to integrate CNNs into translation systems include the work of Kalchbrenner and Blunsom, who designed a hybrid architecture using a CNN as the encoder and an RNN as the decoder (Figure 2.3). This approach leveraged the CNN's ability to extract hierarchical features from the source sentence while relying on the sequential modeling capacity of the RNN to generate fluent target sequences.

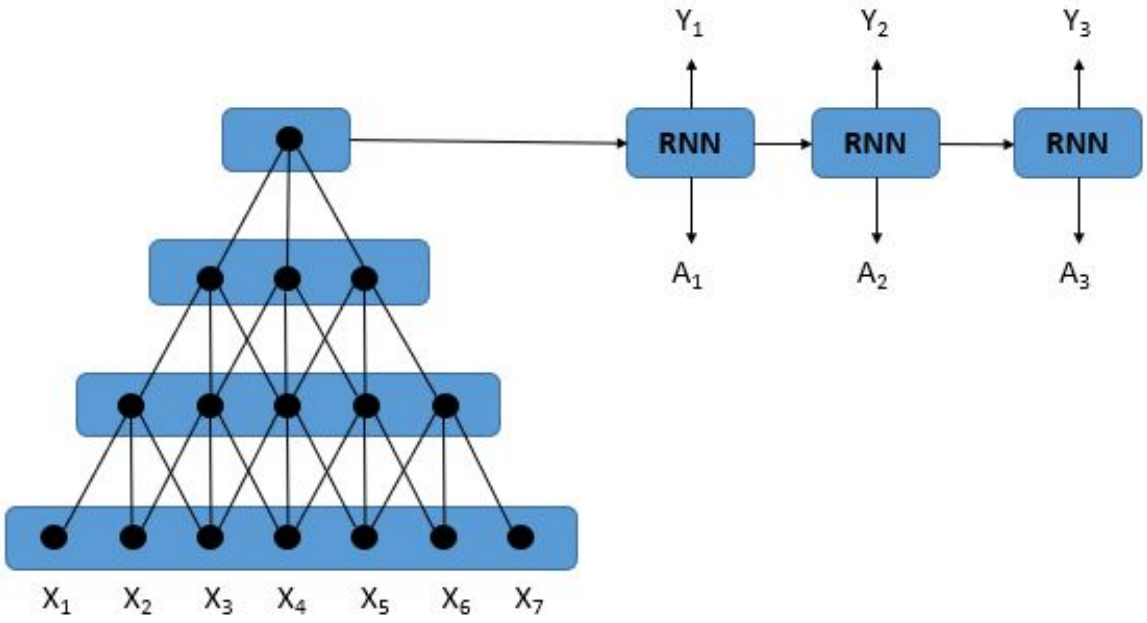


Figure 2.3: Illustration of a hybrid MT model with a CNN encoder and an RNN decoder.

Subsequently, fully CNN-based architectures were proposed to further exploit the parallelism and efficiency of convolutional operations. Łukasz Kaiser and Samy Bengio introduced an Extended Neural GPU adapted to CNNs, achieving competitive results

comparable to RNN-based models [8]. Around the same time, Kalchbrenner et al. developed ByteNet, a character-level CNN model that demonstrated state-of-the-art performance in its domain, although word-level translation quality was initially less competitive [1] (Figure 2.4).

One of the key advantages of CNN-based NMT is training efficiency. Unlike RNNs, which process input sequences token by token, CNNs operate on the entire sentence simultaneously using convolutional filters, enabling faster computation and better handling of gradient vanishing. Nonetheless, CNN models face limitations in translation quality. Since convolutional filters have a fixed receptive field, capturing long-range dependencies requires stacking multiple convolutional layers, which may dilute detailed relational information between distant tokens. Additionally, compressing the source sentence into a fixed representation vector can result in information loss, a challenge shared with RNNs. Attention mechanisms were later introduced to partially mitigate this limitation by allowing the model to focus dynamically on relevant parts of the input during decoding.

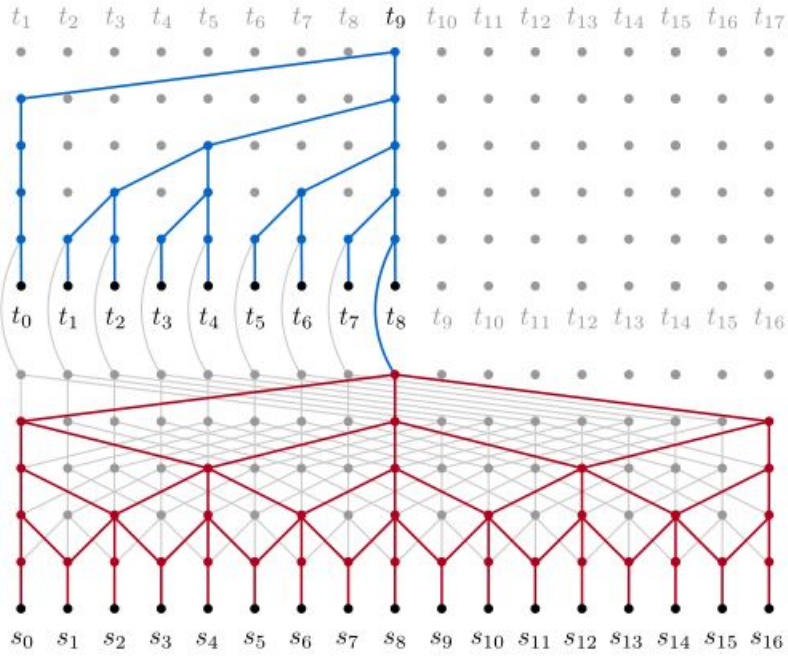


Figure 2.4: Architecture of the ByteNet model. The target decoder (blue) is stacked on top of the source encoder (red). The decoder generates the variable-length target sequence using dynamic unfolding [1].

**Inference Methods**

In practical applications, once an RNN-based translation model is trained, it can be used to generate translations for new, unseen source sentences. During inference, the model has access only to the input sequence and the encoded hidden representations produced by the encoder. The decoder then sequentially produces the target sentence based on this encoded information. Various decoding strategies exist to guide the generation process, with Greedy Search being one of the simplest approaches.

**Greedy Search.** Greedy search operates by selecting the most probable token at each step as the next element of the output sequence. The process proceeds as follows:

- **Initialization:** The decoder receives the encoded representation of the source sentence along with the <SOS> token, marking the start of decoding.
- **Probability Computation:** At each time step, the model predicts a probability distribution over the entire target vocabulary for the next token.
- **Token Selection:** The token with the highest probability is chosen as the next word in the output and is fed back into the decoder for the subsequent time step.
- **Iteration:** Steps 2 and 3 are repeated until the <EOS> token is produced, signaling the end of the generated sentence.

While greedy search is fast and easy to implement, it can be myopic: because it always selects the locally optimal choice, it may fail to produce the most accurate overall translation sequence.

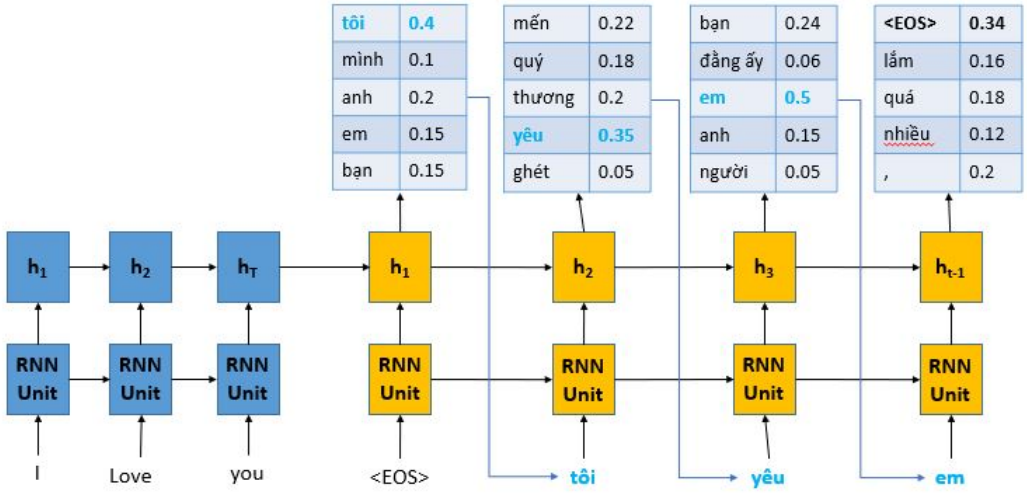


Figure 2.5: Visualization of greedy search decoding applied to an RNN-based NMT model.

### Beam Search

While Greedy Search is straightforward and efficient, it often falls short in capturing the best overall translation when multiple candidate tokens have similar probabilities. To address this limitation, Beam Search offers a more flexible and robust decoding strategy. Originally proposed in [9] for general sequence prediction tasks, Beam Search has become a standard technique in machine translation for generating higher-quality output sequences.

The core idea behind Beam Search is to maintain a set of the top-$n$ most likely partial translations at each decoding step, rather than committing to a single token as in Greedy Search. At each time step, the decoder evaluates all possible next tokens for each candidate sequence in the beam, generating multiple new sequences. From these, only the $n$ sequences with the highest cumulative probabilities are retained for the next step. This process continues iteratively until the <EOS> token is generated in all sequences. Finally, the complete translation is selected from the beam by choosing the sequence with the highest total probability.

By exploring multiple promising candidates simultaneously, Beam Search reduces the risk of early commitment to suboptimal tokens, allowing for more accurate and

contextually appropriate translations. However, this improvement comes at a computational cost: maintaining multiple sequences increases memory usage and processing time. To balance efficiency and performance, the beam width ($n$) is typically kept moderate; studies [10] suggest that values between 5 and 10 provide a good trade-off between translation quality and decoding speed.

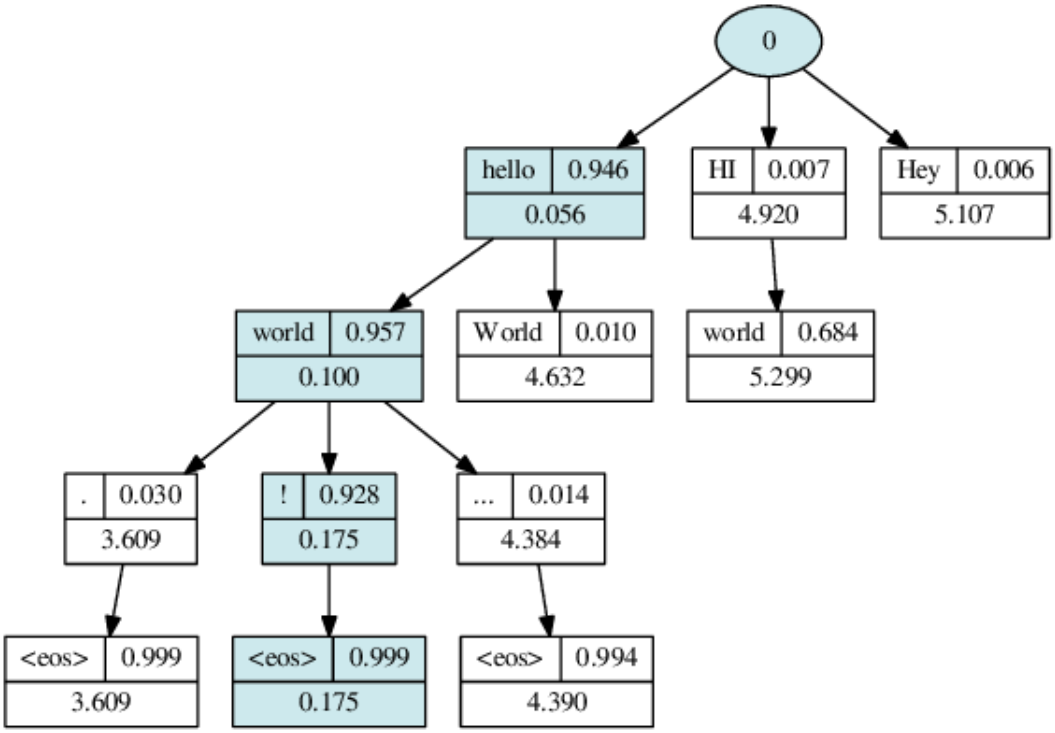


Figure 2.6: Illustration of the Beam Search decoding process. Multiple candidate sequences are explored at each step, and the sequence with the highest total probability is selected as the final output.

### Encoder-Decoder Architecture

The Encoder-Decoder framework has become a foundational paradigm in machine translation and sequence-to-sequence modeling. Originally introduced by Kalchbrenner and Blunsom [11], this design separates the translation process into two distinct yet interconnected components: an Encoder, which transforms a source sentence into a dense, semantic representation, and a Decoder, which generates the corresponding sentence in the target language from this representation. While numerous adaptations have been proposed—ranging from variations in layer types to specialized attention mechanisms—the core principle of encoding semantic information and decoding it into a new sequence has remained consistent. Early implementations experimented with heterogeneous architectures, such as combining a CNN-based encoder with an RNN-based decoder [11], highlighting the flexibility of the framework. Conceptually, this architecture can be viewed as an end-to-end mapping from one language's semantic space to another, producing translations without requiring explicit alignment rules.

In the Transformer model, both the Encoder and Decoder consist of stacks of identical layers (six layers for each, as in [12]). Each encoder layer contains two primary components: a multi-head self-attention mechanism and a fully connected feed-forward network. To stabilize training, each sub-layer is wrapped with residual connections

followed by layer normalization, such that for an input $x$ and a sub-layer function SubLayer($x$), the output is computed as:

$$\text{LayerNorm}(x + \text{SubLayer}(x))$$

The encoder processes the source sentence using self-attention to capture long-range dependencies and produce a final representation rich in semantic context.

Decoder layers extend the encoder design by including an additional multi-head self-attention sub-layer that attends to previously generated tokens, allowing the model to maintain sequential coherence during generation. Notably, positional information is handled differently in the decoder: while input embeddings are augmented with positional encodings to retain token order, the decoder's self-attention prevents future tokens from influencing the current output, ensuring causality in generation.

Positional encoding in the Transformer serves to inject explicit information about token positions into the embeddings. For an embedding vector of length $n$ and token position $i$, the encoding is defined as:

$$P_{i,2j} = \sin\left(\frac{i}{10000^{\frac{2j}{n}}}\right), \quad P_{i,2j+1} = \cos\left(\frac{i}{10000^{\frac{2j+1}{n}}}\right)$$

for $0 \leq j < n/2$, allowing the model to distinguish relative and absolute positions of tokens in a sentence [12] . Figure 2.7 illustrates how the positional encoding matrix is constructed and integrated with token embeddings to form input representations for the Transformer encoder and decoder.

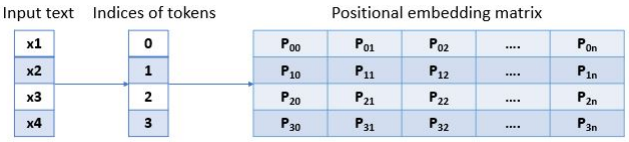


Figure 2.7: Visualization of the positional encoding scheme. Each row corresponds to a token position in the input, and each column represents a dimension in the embedding space.

### Self-Attention Mechanism

Self-attention is a cornerstone of the Transformer architecture, serving as the primary method for capturing dependencies within a sequence. Unlike earlier attention mechanisms proposed by Bahdanau et al. [5] and Luong et al. [7], which focus on aligning encoder and decoder states, self-attention allows each token in a sequence to directly interact with every other token, enabling the model to understand both local and global context simultaneously. This capability makes it particularly effective for representing long-range relationships and is widely recognized as a key innovation behind the Transformer's success.

When an input sequence $X$ of length $T$ is fed into a self-attention layer, it is first transformed into three separate matrices: Queries ($Q^h$), Keys ($K^h$), and Values ($V^h$), each derived by multiplying $X$ with learnable weight matrices $W_Q^h$, $W_K^h$, and $W_V^h$, respectively. The attention mechanism computes a relevance score between each query and all keys, normalizes the scores with a softmax function, and uses these weights to

generate a weighted sum of the values. This produces the context vector $C^h$, which encodes relationships between tokens across the sequence:

$$C^h = V^h \cdot \text{softmax}\left(\frac{Q^h K^{hT}}{\sqrt{d}}\right) \tag{2.4}$$

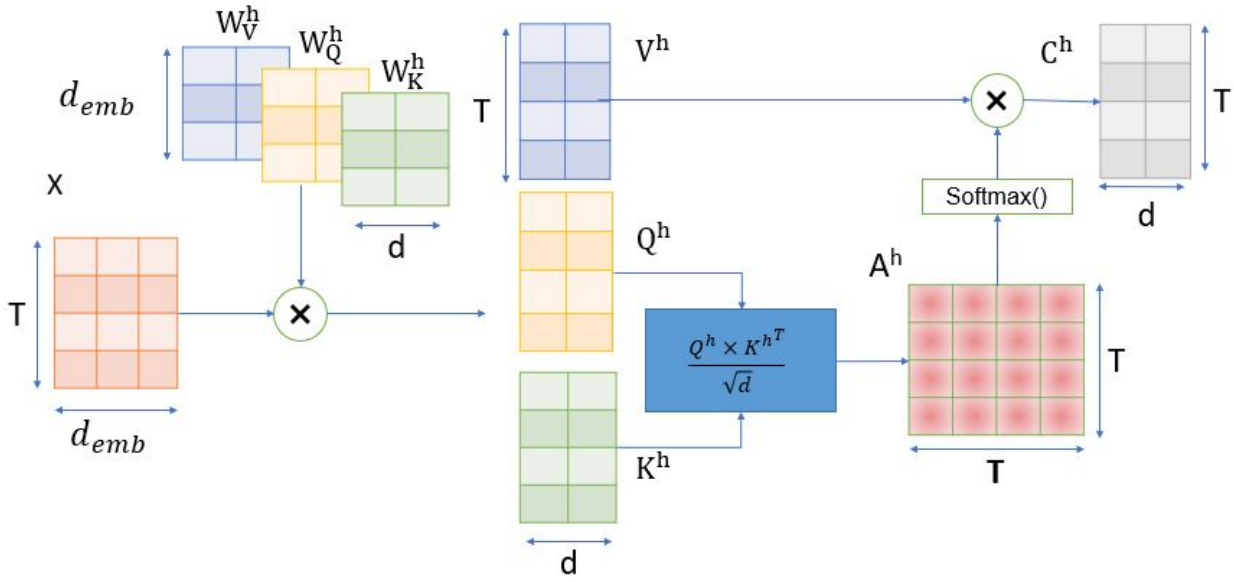


Figure 2.8: Illustration of self-attention, showing how each token attends to all others to generate context-aware representations.

To increase the model's representational capacity, the Transformer employs multi-head attention. Here, multiple independent attention mechanisms—referred to as heads—process the input in parallel, each producing its own context vector. These vectors capture complementary aspects of the sequence, and their concatenation forms the final representation, enabling the model to focus on different positions and semantic subspaces simultaneously. The original Transformer uses eight attention heads [12], though this number can be adjusted according to model requirements. Multi-head attention is depicted in Figure 2.9.

**Notation:**

- $X$: Input embeddings
- $T$: Length of the input sequence
- $d_{emb}$: Dimensionality of embedding vectors
- $d$: Hidden representation size
- $Q^h, K^h, V^h$: Query, Key, and Value matrices for head $h$
- $W_Q^h, W_K^h, W_V^h$: Learnable weight matrices for the corresponding projections
- softmax(): Softmax function for normalizing attention scores

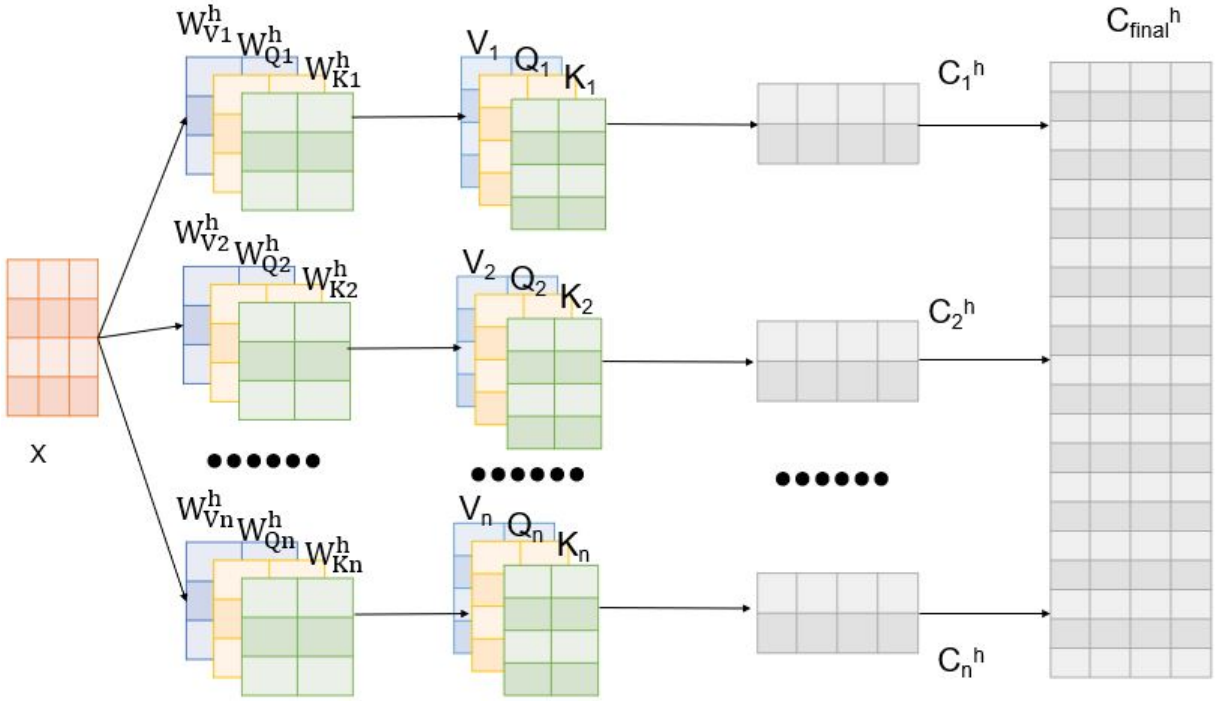


Figure 2.9: Multi-head attention in Transformers. Each head independently computes context vectors, which are then concatenated to produce the final output.

- $A^h$: Attention weight matrix for head $h$
- $C^h$: Context vector for head $h$

## 2.2 Reinforcement Learning

Reinforcement learning (RL) studies how goal-directed agents improve behavior by interacting with an environment and receiving evaluative feedback in the form of rewards. The standard mathematical model is a Markov decision process (MDP) $(\mathcal{S}, \mathcal{A}, p, r, \gamma)$ with a (possibly large or continuous) state space $\mathcal{S}$, action space $\mathcal{A}$, transition kernel $p(s_{t+1} \mid s_t, a_t)$, reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, and discount factor $\gamma \in (0, 1]$ that downweights distant outcomes. A (potentially stochastic) policy $\pi_\theta(a \mid s)$ induces trajectories $\tau = (s_0, a_0, s_1, a_1, \dots)$ and an expected discounted return

$$J(\theta) = \mathbb{E}_{\tau \sim p_\theta}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right], \quad p_\theta(\tau) = p(s_0) \prod_{t \geq 0} \pi_\theta(a_t \mid s_t)\, p(s_{t+1} \mid s_t, a_t). \tag{2.5}$$

The agent's learning problem is to choose parameters $\theta$ that maximize $J(\theta)$ using data it gathers while acting.

**Why RL is hard.** Compared to supervised learning, RL faces two core difficulties. *Credit assignment* means that the effect of an action may be delayed: outcomes many time steps later must be attributed to earlier decisions. *Distribution shift* means that as the policy changes, the data distribution changes too; naively applying gradient updates can push the policy into regions where previous estimates are inaccurate, causing

instability. In addition, practical deep RL contends with function approximation errors, partial observability (POMDPs), exploration–exploitation trade-offs, and reward sparsity or misspecification.

**Two broad families (context).** Although this section focuses on policy optimization, it is helpful to situate it among classical families. *Value-based* methods approximate $Q^\pi(s, a)$ (or its optimal counterpart) and act greedily with respect to the learned value; this shines in discrete, low-dimensional action spaces. *Policy-based* methods directly optimize a parameterized policy, often with an auxiliary value function (actor–critic) for variance reduction. Policy methods naturally support stochastic policies and continuous-action distributions, but typically require additional mechanisms to keep updates stable. The remainder of this section provides conceptual background on policy gradients, then presents two stabilization strategies-trust regions and proximal objectives and finally a rank-based variant useful when reward scales are heterogeneous.

### 2.2.1 Policy-based Gradient Methods

**Key idea and historical intuition.** Policy-gradient (PG) methods adjust the probability of sampled actions in proportion to how good those actions turned out to be. If an action improved return, the method slightly increases its log-probability under the current policy; if it harmed return, the method decreases it. This direct manipulation of the policy sidesteps the need to derive a policy from a value table and extends cleanly to complex action parameterizations.

**The policy-gradient theorem.** The central identity is

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}\big[\nabla_\theta \log \pi_\theta(a_t \mid s_t)\, A^\pi(s_t, a_t)\big], \quad A^\pi(s, a) = Q^\pi(s, a) - V^\pi(s), \tag{2.6}$$

which states that the gradient of expected return equals an expectation of score functions weighted by (advantage) returns. Two important consequences are (i) *baseline invariance*: subtracting any function of state, $b(s_t)$, from $Q^\pi$ leaves the gradient unbiased and can reduce variance, and (ii) *on-policy sampling*: the expectation is taken under the current policy's trajectory distribution.

**From REINFORCE to actor–critic.** REINFORCE[13] uses Monte Carlo returns, yielding an unbiased but high-variance estimator. Actor–critic methods add a learned baseline $V_\phi(s) \approx V^\pi(s)$, replacing raw returns by *advantages* $\hat{A}_t$ (e.g., temporal-difference errors) to reduce variance[14]. A widely used construction is *generalized advantage estimation* (GAE)[15]:

$$\hat{A}_t^{\mathrm{GAE}(\gamma,\lambda)} = \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_{t+l}, \qquad \delta_t = r_t + \gamma V_\phi(s_{t+1}) - V_\phi(s_t), \tag{2.7}$$

with $\lambda \in [0, 1]$ controlling the bias–variance trade-off: smaller $\lambda$ uses shorter TD targets (lower variance, higher bias), larger $\lambda$ uses longer returns (higher variance, lower bias). In practice, advantages are normalized (zero mean, unit variance) per batch, and an *entropy bonus* encourages the policy to maintain exploration[16].

**Continuous actions and parameterizations.** Policy-based methods handle continuous control by parameterizing a distribution over actions, commonly a diagonal Gaussian, $\pi_\theta(\cdot \mid s) = \mathcal{N}(\mu_\theta(s), \mathrm{diag}(\sigma_\theta^2(s)))$, often learning $\log \sigma$ for stability. For bounded actions, tanh-squashed Gaussians or Beta distributions are typical[17][18]. In discrete spaces, a softmax over logits provides a flexible mixed strategy.

**What goes wrong without constraints.** Naïve first-order updates can *overshoot*: a large step may move the policy to regions unsupported by the data that produced the gradient estimate. This can lead to *policy collapse* (premature determinism), brittle performance under reward rescaling, or catastrophic regressions between iterations. These issues motivate mechanisms that explicitly limit policy movement per update[19].

**Exploration, partial observability, and baselines (background notes).** Stochastic policies serve dual roles: they (i) provide the gradient signal via $\nabla \log \pi$, and (ii) implement exploration. Under partial observability, policies may be recurrent (e.g., GRUs) to carry information across time. Baselines can be state-dependent (value functions) or action-dependent (Q-baselines in compatible function approximation), and reward/advantage normalization helps when reward scales vary across tasks.

### 2.2.2 Trust Region Methods

**Stability via conservative updates.** Trust regions formalize the intuition that a policy should change only so much per update, balancing two pressures: updates must be large enough to make progress, yet small enough that the first-order surrogate remains predictive of true improvement. The divergence between old and new policies is measured by the Kullback–Leibler (KL) divergence, averaged over states visited by the old policy.

**TRPO in words.** Trust Region Policy Optimization (TRPO)[20] maximizes a first-order surrogate of expected return while enforcing a constraint on the average change in policy:

$$\max_\theta \ \mathbb{E}_t\big[\rho_t(\theta)\,\hat{A}_t\big] \quad \text{s.t.} \quad \mathbb{E}_t[D_{\mathrm{KL}}(\pi_{\theta_{\mathrm{old}}}(\cdot \mid s_t) \,\|\, \pi_\theta(\cdot \mid s_t))] \le \delta, \quad \rho_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\mathrm{old}}}(a_t \mid s_t)}. \tag{2.8}$$

Intuitively, the constraint restricts how much probability mass can be reallocated from the old policy to the new one at states we actually visited, maintaining the trustworthiness of the surrogate objective.

**Natural gradient geometry (conceptual background).** The KL constraint induces a Riemannian metric on the policy manifold with Fisher information matrix $F = \mathbb{E}[\nabla \log \pi \, \nabla \log \pi^\top]$. Taking a step proportional to $F^{-1}g$ (where $g$ is the policy-gradient of the surrogate) yields a *natural gradient* step that respects the information geometry of the parameterization: equal KL movement in any direction corresponds to equal "length". TRPO approximates $F^{-1}g$ with conjugate gradients and scales the step via a backtracking line search to satisfy the KL bound and to ensure the surrogate improves[21][22].

**Why TRPO helps (background insights).**

- **Monotonic improvement intuition.** Under mild assumptions, constraining the KL ensures the expected return of the new policy is not much worse than the surrogate predicts, supporting near-monotonic improvement across iterations.
- **Reward-scale robustness.** Because the constraint is geometric (in probability space), TRPO is less sensitive to arbitrary rescalings of reward magnitudes than plain policy gradients.
- **Distribution shift control.** By keeping the policy close, the state distribution under the new policy remains close to that of the data-generating policy, so first-order estimates remain valid.

**Trade-offs and practicalities.** TRPO introduces second-order machinery (Fisher–vector products, line searches), which increases engineering complexity and wall-clock time per update. It is typically used with large on-policy batches, GAE advantages, value-function fitting, and entropy regularization. Hyperparameters include the KL radius $\delta$ (smaller means more conservative steps), CG iterations/tolerance, and line-search parameters. Despite its robustness, the added complexity motivated simpler, first-order methods that preserve the trust-region spirit.

### 2.2.3 Proximal Policy Optimization Algorithms

**Design goal and intuition.** Proximal Policy Optimization (PPO)[23] aims for TRPO-like stability without second-order optimization. The core idea is to *softly* limit the policy update by shaping the surrogate objective so that extreme importance ratios are penalized or clipped, thus forming an implicit proximal region around the old policy.

**The clipped surrogate (background).** Define $\rho_t(\theta) = \pi_\theta(a_t \mid s_t)/\pi_{\theta_{\text{old}}}(a_t \mid s_t)$. The *clipped* objective is

$$L^{\text{CLIP}}(\theta) = \mathbb{E}_t\Big[\min\Big(\rho_t(\theta)\,\hat{A}_t,\ \text{clip}(\rho_t(\theta),\,1-\epsilon,\,1+\epsilon)\,\hat{A}_t\Big)\Big]. \tag{2.9}$$

When $\hat{A}_t > 0$ and $\rho_t > 1+\epsilon$, the clipped term prevents the update from excessively increasing the probability of advantageous actions; when $\hat{A}_t < 0$ and $\rho_t < 1-\epsilon$, it prevents overly drastic decreases. This yields a plateau-like loss that discourages destructive overshoots while allowing improvements within a safe corridor.

**Composite training objective.** PPO optimizes a combined loss including (i) the clipped policy loss, (ii) a value-function regression loss with targets $\hat{R}_t$ (e.g., GAE-lambda returns), and (iii) an entropy bonus to sustain exploration:

$$L(\theta,\phi) = \mathbb{E}\left[L^{\text{CLIP}}(\theta) - c_V\,(V_\phi(s_t) - \hat{R}_t)^2 + c_H\,\mathcal{H}(\pi_\theta(\cdot \mid s_t))\right]. \tag{2.10}$$

Training proceeds with first-order optimizers (e.g., Adam), multiple epochs over each on-policy batch, minibatch shuffling, advantage normalization, and optional early stopping if the measured KL between old and new policies exceeds a threshold.

**Advantages of PPO**

- **Simplicity.** No second-order solvers or line searches are required; implementation is compact and hardware-friendly.
- **Robustness.** Clipping provides a strong safety mechanism against pathological updates arising from noisy advantages or outliers.
- **General applicability.** PPO has strong empirical performance across discrete and continuous control tasks and serves as a reliable baseline.

The clipping introduces bias relative to the unconstrained surrogate and can under-update when advantages are large, but in practice the bias–variance trade-off is usually favorable.

**Background notes: hyperparameters and diagnostics.** Typical settings include $\epsilon \in [0.1, 0.3]$, GAE $\lambda \in [0.9, 0.97]$, value loss weight $c_V$ tuned so that value errors do not dominate, and a small but nonzero entropy weight $c_H$. Practical diagnostics monitor empirical KL, explained variance of the value function, and the fraction of samples where the objective is clipped (a proxy for how frequently updates hit the proximal boundary).

**Pitfalls to recognize.** Coupling the policy and value losses can cause interference; separate learning rates or decoupled optimizers sometimes help. Excessive epochs over the same on-policy data can overfit; early stopping on KL or performance is a common safeguard. For sparse rewards, auxiliary shaping or curiosity-driven exploration may still be needed.

### 2.2.4 Group Relative Policy Optimization

**Motivation: heterogeneous or uncalibrated rewards.** In many modern applications, especially sequence generation and preference learning, the reward for a candidate may be a composite of several signals (e.g., fluency, factuality, safety) with arbitrary scales and noise properties. Tuning absolute weights and calibrations can dominate engineering effort. *Group Relative Policy Optimization* (GRPO)[24] addresses this by making the learning signal *scale-invariant*: it depends on the *relative ordering* of candidates generated for the same input, not on their raw magnitudes.

**Mechanism in words.** For each input or state $x$, sample a small group of $K$ candidates from the current policy, score each candidate with a scalar evaluator (which may itself be a composite), and produce a ranking of the group from best to worst. Replace magnitude-based advantages by centered, rank-derived advantages within the group: top-ranked items receive positive credit, bottom-ranked negative credit, and the group sums to zero. These advantages then plug directly into the familiar PPO clipped objective, preserving its practical safeguards.

**Constructing rank-based advantages.** There are several principled ways to form a zero-mean, scale-invariant signal:

$$\begin{aligned}
\text{Centered rank:} \quad & \tilde{A}_j = \tau \, \frac{2\,\mathrm{rank}(j) - (K+1)}{K-1}, \quad \sum_{j=1}^{K} \tilde{A}_j = 0, \\
\text{Pairwise wins–losses (Borda):} \quad & \tilde{A}_j \propto \sum_{m \neq j} \big( \mathbb{1}\{R_j > R_m\} - \mathbb{1}\{R_j < R_m\} \big), \\
\text{Smooth pairwise (Bradley–Terry):} \quad & \tilde{A}_j \propto \sum_{m \neq j} \big( \sigma(R_j - R_m) - \tfrac{1}{2} \big),
\end{aligned}$$

where $\mathrm{rank}(j) \in \{1, \ldots, K\}$ orders candidates by their evaluator score $R_j$ (higher is better), $\sigma$ is a logistic, and $\tau > 0$ controls sharpness. The policy update mirrors PPO with $\hat{A} \leftarrow \tilde{A}$:

$$L^{\mathrm{GRPO}}(\theta) = \mathbb{E}_{(x,a_j)\in\mathcal{G}} \Big[ \min \big( \rho_j(\theta)\, \tilde{A}_j,\ \mathrm{clip}(\rho_j(\theta), 1-\epsilon, 1+\epsilon)\, \tilde{A}_j \big) \Big], \quad \rho_j(\theta) = \frac{\pi_\theta(a_j \mid x)}{\pi_{\theta_{\mathrm{old}}}(a_j \mid x)}. \tag{2.11}$$

**Why rankings help?.**

- **Scale invariance.** Any affine transformation $R'_j = aR_j + b$ with $a > 0$ preserves the ordering and thus the learning signal, reducing sensitivity to reward engineering and enabling heterogeneous evaluators to be combined by simple fusion followed by ranking.
- **Per-input centering.** Because the advantages sum to zero within each group, updates are naturally centered at the input level, stabilizing gradients and reducing sensitivity to batch composition.
- **Top-heavy credit.** Ranking emphasizes relative improvements among the best candidates, which often matter most for perceived quality in generative tasks.

**Design levers and background considerations.**

1. **Group size $K$.** Larger $K$ yields finer rank resolution but increases sampling/evaluation cost. Choices like $K \in \{4, 8, 16\}$ are common; the sweet spot depends on evaluator latency and the policy's diversity.
2. **Fused scoring.** If multiple evaluators exist, combine them via a weighted sum, a learned scalarizer, or lexicographic rules before ranking. Auditing per-aspect win rates helps detect domination by one dimension.
3. **Ties and noise.** Deterministic tie-break rules (e.g., by log-probability) avoid label noise; smooth pairwise constructions (Bradley–Terry) reduce brittleness under noisy scores.
4. **Temperature $\tau$.** Controls how aggressively top ranks are rewarded; lower $\tau$ spreads credit more evenly, higher $\tau$ focuses it on the top few.

**Scope, strengths, and caveats.** GRPO is well suited to settings where "which candidate is better" is easier to judge than "by how much", such as preference modeling or composite quality metrics. When absolute magnitudes encode crucial risk/cost information, purely rank-based signals may discard useful calibration; hybrids that mix rank- and magnitude-based advantages can recover this information (e.g., convex combinations of centered ranks and z-scored rewards). As with PPO, GRPO is largely on-policy: reusing stale groups without correction can bias learning.

**Background pseudocode (for intuition).**

```
for iteration = 1..T:
  collect batches of inputs x
  for each x:
    sample K candidates a_1..a_K ~ pi_old(.|x)
    evaluate scores R_1..R_K (possibly composite)
    compute ranks and centered rank advantages ~A_1..~A_K
  form dataset D = {(x,a_j,~A_j, logp_old)}
  for epoch = 1..E:
    for minibatch in shuffle(D):
      compute rho = exp(logp_new - logp_old)
      policy_loss = mean(min(rho*~A, clip(rho,1-eps,1+eps)*~A))
      value_loss  = mse(V(s), targets)  # optional
      entropy     = H(pi_new(.|s))
      optimize(policy_loss - c_V*value_loss + c_H*entropy)
  set pi_old <- pi_new
```

**Connections.** GRPO's use of ranks relates conceptually to listwise learning-to-rank surrogates. It also resembles preference-based RL where pairwise comparisons define a Bradley–Terry likelihood over returns; GRPO differs by keeping a standard policy-gradient backbone with proximal updates and by using ranks as an advantage signal rather than a direct likelihood term.

**Summary of background themes.** Policy-gradient methods provide a direct route to optimizing complex, stochastic policies but require care to remain stable. Trust-region approaches (TRPO) use a geometric KL constraint to ensure conservative, reliable improvement; proximal methods (PPO) approximate this with a first-order, clipped surrogate that is simple and robust. When reward scales are heterogeneous or noisy, GRPO replaces magnitude-sensitive advantages with rank-based, per-input signals while retaining PPO's practical machinery. Across all methods, value baselines (actor–critic), advantage normalization, entropy regularization, and careful diagnostics (e.g., empirical KL, value explained variance, clip fraction) are central to stable training in practice.

## 2.3 Scoring Metrics

This section surveys the families of metrics used to score system outputs $\hat{y}$ against references $y^\star$ (or against implicit human preferences). We group them by the type of evidence they exploit: surface-level *lexical overlap*, *semantic* similarity in embedding or model space, and *syntactic* structure captured as graphs. We then discuss *graph similarity* measures that compare structured representations directly. Throughout, we

highlight what each metric measures, typical formulations, and practical considerations when turning them into training rewards.

### 2.3.1 Lexical Matching Metrics

**What they measure.** Lexical metrics quantify $n$-gram or character overlap between a hypothesis $\hat{y}$ and a reference $y^\star$. They reward local phrase reuse and penalize omissions, with weak sensitivity to paraphrase.

**BLEU.[25]** BLEU aggregates modified $n$-gram precisions with a brevity penalty (BP) to discourage overly short hypotheses:

$$\mathrm{BLEU} = \mathrm{BP} \cdot \exp\Big(\sum_{n=1}^{N} w_n \log p_n\Big), \qquad \mathrm{BP} = \begin{cases} 1 & \text{if } |\hat{y}| \geq |y^\star|, \\ \exp(1 - |y^\star|/|\hat{y}|) & \text{otherwise,} \end{cases} \tag{2.12}$$

where $p_n$ is the clipped precision of $n$-grams and $\{w_n\}$ are typically uniform over $n = 1{:}4$. Common refinements include smoothing of $p_n$ for rare $n$-grams and sentence-level BLEU for per-instance rewards.

**METEOR (lexico-semantic blend)[26].** METEOR computes unigram precision/recall with stemming/synonym matching and fragmentation penalties. While it uses lexical units, its alignment heuristic gives it modest paraphrase tolerance relative to pure $n$-gram counts.

**Strengths and caveats.** Lexical metrics are simple, fast, and correlate with adequacy when phrasing is constrained. They struggle with legitimate paraphrases and meaning preservation under rewording. As rewards, they can over-constrain style unless complemented by semantics.

### 2.3.2 Semantic Matching Metrics

**What they measure.** Semantic metrics compare *meaning* rather than surface form, typically by aligning contextual embeddings or by using learned quality estimators trained on human judgments.

**BERTScore.[27]** Tokens (or subwords) are embedded with a pretrained encoder; similarity is the (greedy or optimal) alignment of hypothesis to reference tokens under cosine similarity:

$$\mathrm{P} = \frac{1}{|\hat{y}|} \sum_{i \in \hat{y}} \max_{j \in y^\star} \cos(e_i, e'_j), \quad \mathrm{R} = \frac{1}{|y^\star|} \sum_{j \in y^\star} \max_{i \in \hat{y}} \cos(e_i, e'_j), \quad \mathrm{F}_\beta = \frac{(1+\beta^2)\,\mathrm{P}\,\mathrm{R}}{\beta^2\,\mathrm{P} + \mathrm{R}}. \tag{2.13}$$

Layer choice, IDF weighting, and rescaling against references influence robustness.

**MoverScore / Word Mover's Distance.[28]** Treat token embeddings as distributions and compute the minimum transport cost (Earth Mover's Distance, EMD) between $\hat{y}$ and $y^\star$:

$$\mathrm{MS}(\hat{y}, y^\star) = -\min_{T \in \Pi(\mathbf{p}, \mathbf{q})} \sum_{i,j} T_{ij}\, \mathrm{cost}(e_i, e'_j), \tag{2.14}$$

where $\Pi(\mathbf{p}, \mathbf{q})$ are couplings between token masses and cost is typically $1 - \cos(\cdot, \cdot)$. This captures many-to-many soft alignments.

**Learned quality estimators COMET[29].** Neural regressors encode $(x, \hat{y}, y^\star)$ (or $(x,$
*haty*) for reference-free QE) and predict a human-like quality score. They deliver strong system-level correlation but introduce domain dependence and require careful calibration if used as rewards.

**Strengths and caveats.** Semantic metrics are robust to paraphrase and better track adequacy/fluency trade-offs. They can be sensitive to the underlying encoder's biases and are heavier to compute. As training signals, they benefit from normalization and outlier clipping.

### 2.3.3 Syntactic Graphs

**Why syntax.** Meaning often hinges on *relations*—who did what to whom, what was negated, modified, or quantified. Syntactic graphs provide a structured view of these relations that complements lexical and semantic scores.

**Dependency graphs.[30]** A dependency parse is a labeled directed graph $G = (V, E, \ell)$ with one node per token and edges $(h \to d,\ \ell_{hd})$ encoding head–dependent relations from a tagset (e.g., Universal Dependencies). Let $G_{\text{src}}$ and $G_{\text{hyp}}$ denote source and hypothesis parses (or $G_{\text{ref}}$ for the reference). Around an entity or keyword span, we can extract a $k$-hop *ego subgraph* $G^{(k)}$ to localize evaluation to critical neighborhoods (e.g., dosage, negation, qualifiers).

**Practical notes.** Parsing quality affects any syntax-based score; domain-mismatch parsers can introduce noise. When used as rewards, restrict comparisons to high-confidence regions (e.g., within $k$ hops of entities) and prefer label sets that are stable across languages or versions.

### 2.3.4 Graph Similarity Metrics

**What they measure.** Graph similarity metrics compare structure directly, either *exactly* through edit/matching objectives or *approximately* through kernels and spectral/embedding proxies. They are useful when correctness is tied to preserving relational patterns (dependencies, core roles, modifier scope).

**Graph edit distance (GED)[31].** GED is the minimum cost of transforming $G_1$ into $G_2$ via node/edge substitutions, insertions, and deletions:

$$\mathrm{GED}(G_1, G_2) = \min_{\pi \in \mathcal{M}} \Big( \sum_{v \in V_1} c_v(v, \pi(v)) + \sum_{e \in E_1} c_e(e, \pi(e)) \Big), \tag{2.15}$$

over (partial) node mappings $\pi$ with label-aware costs $c_v, c_e$. Exact GED is NP-hard; practical use relies on heuristics (A* bounds, bipartite matching relaxations) and is best suited to small subgraphs (e.g., entity neighborhoods).

**Graph kernels.** Kernels provide positive-definite similarity without explicit alignment. Two widely used families:

- **Weisfeiler–Lehman (WL) subtree kernel[32]:** iteratively relabel nodes by hashing their label and multiset of neighbor labels up to height $h$, then compare label-count histograms. Captures nested neighborhood patterns; efficient for large graphs.
- **Random walk / shortest-path kernels[33]:** compare counts of label-consistent walks or shortest paths between graphs; sensitive to cycles and path labels, with closed forms via matrix operations.

Kernels naturally support SVMs or can be normalized into $[0, 1]$ scores.

**Spectral and embedding similarities.** Compute graph embeddings and compare with vector distances:

$$\mathrm{spec}(G_1, G_2) = \left\| \lambda(L_{G_1}) - \lambda(L_{G_2}) \right\|_p, \tag{2.16}$$

where $\lambda(L_G)$ lists (padded) eigenvalues of the (normalized) Laplacian. Alternatively, embed graphs or subgraphs via GNNs or graph2vec and use cosine similarity. Spectral scores capture global shape; embedding scores trade interpretability for speed and differentiability.

**Local, entity-centered graph similarity.** For information-sensitive evaluation, restrict to a $k$-hop subgraph $G^{(k)}$ around an entity span $\mathcal{E}$. A simple neighborhood-overlap score is

$$R^{(k)}_{\mathrm{dep}} = \frac{1}{|\mathcal{E}|} \sum_{e \in \mathcal{E}} \frac{|\,\mathcal{N}^{(k)}_{G_{\mathrm{hyp}}}(e) \cap \mathcal{N}^{(k)}_{G_{\mathrm{ref}}}(e)\,|}{|\,\mathcal{N}^{(k)}_{G_{\mathrm{hyp}}}(e) \cup \mathcal{N}^{(k)}_{G_{\mathrm{ref}}}(e)\,|}, \tag{2.17}$$

optionally label-aware (edge labels, directions) or replaced by a WL-kernel restricted to $G^{(k)}$. This focuses the metric on preserving roles (negation, dosage, attributions) most critical to meaning.

**Strengths and caveats.** Graph metrics make relation errors visible and quantifiable, complementing lexical/semantic scores. They require stable parsing/graphing and careful alignment between graphs (token spans, node anchors). Exact methods are costly on large graphs; neighborhood restriction, kernels, or embeddings provide tractable surrogates. As rewards, graph similarities benefit from smoothing (e.g., averaging over $k$ and label subsets) to reduce brittleness.

**Putting the families together.** In practice, robust evaluation (and reward design) blends signals: lexical metrics for local fidelity and omission control; semantic metrics for paraphrase-tolerant adequacy; syntactic/graph metrics for relational correctness. Composite scores can be formed by weighted sums with per-component normalization, or by rank-based aggregation to avoid scale calibration issues.

## 2.4 Summary of the Chapter

This chapter surveys the foundations needed for later work. It traces machine translation from rule-based systems to statistical methods and then to neural approaches, detailing RNN- and CNN-based NMT, attention mechanisms (Bahdanau/Luong), decoding strategies (greedy, beam), and the encoder–decoder/Transformer toolkit (positional encoding, self- and multi-head attention). It then outlines reinforcement learning basics: the MDP formalism, policy gradients (REINFORCE, actor–critic, GAE), and stability mechanisms via trust regions (TRPO), proximal objectives (PPO), and rank-based Group Relative Policy Optimization (GRPO). Finally, it catalogs scoring families used to evaluate or reward translations: lexical overlap (BLEU, chrF, METEOR), semantic similarity (e.g., BERTScore, MoverScore, learned QE), and structure-aware metrics based on dependency/constituency/AMR graphs with graph distances (GED, MCS), kernels (WL), and spectral/embedding similarities, including entity-centered local comparisons. Together, these preliminaries define the concepts, notation, and evaluation tools used throughout the thesis.

# Chapter 3

# Multi-task Learning Entity-Aware Machine Translation

This chapter aims to provide an insight into entity-aware machine translation and the application of entity-aware machine translation to improve the performance of terminology translation in a terminology-sensitive domain. In a terminology-sensitive domain text, the terminologies can be seen as the specific entities among the general tokens. Therefore, a machine translation system that translates entities with high accuracy can assure the performance of terminology translation in the terminology-sensitive domain.

## 3.1 Related Works

Entity-aware machine translation (AEMT) encompasses a family of techniques aimed at overcoming one of the persistent weaknesses of neural machine translation systems: their tendency to mishandle named entities, technical terms, and other semantically critical expressions. Whereas conventional MT models treat all tokens uniformly, AEMT methods intentionally foreground entities by integrating components such as entity recognition modules, terminology constraints, or external knowledge sources. This explicit emphasis on entity fidelity is crucial in high-stakes domains—legal, biomedical, technical—where even minor mistakes in terminology can distort meaning, compromise factual accuracy, or degrade user trust.

A variety of strategies have been explored to achieve this goal. Modrzejewski et al. [34] enhance MT performance by injecting linguistic annotations directly into the model's input. By marking entity spans and types, their method enables the translation system to treat named entities as structured units rather than surface tokens. As a result, the model reduces common errors such as substitution or omission, achieving more reliable entity translation across several language pairs.

Building on similar motivations, Xie et al. [35] introduce an end-to-end design that embeds entity-awareness into both the encoder and decoder. Their encoder incorporates dedicated representations for entity boundaries and categories, while their decoder leverages these enriched features to guide the generation process. This dual-modular approach leads to substantial reductions in entity-level errors and consistent gains over strong baseline NMT systems.

In another line of work, Zeng et al. [36] propose the Extract and Attend (E&A) framework, which explicitly extracts named entities from the source text before trans-

lation. The decoder then uses a specialized attention mechanism to reference these extracted entities during generation. By spotlighting entities throughout the translation process, E&A improves the handling of rare, ambiguous, or domain-specific terms without compromising overall fluency.

Jauhari et al. [37] extend entity-aware techniques to a more structured setting: the joint translation of user queries and their semantic parses. Their model synchronizes entity handling across both natural language and its associated formal representation, mitigating inconsistencies that arise when each output is generated independently. The approach yields improvements in both translation accuracy and semantic correctness, underscoring the broad applicability of entity-aware methodologies.

Finally, Hu et al. [38] explore the potential of large language models as unified multilingual translators through their GenTranslate framework. Rather than relying on specialized MT architectures, their method leverages the generative abilities of LLMs to perform translation directly over text and speech. Crucially, LLMs in this setting demonstrate a strong capacity for preserving entities and reasoning over multilingual context, achieving performance on par with or surpassing traditional translation systems.

## 3.2 The SemEval 2025 dataset

SemEval (Semantic Evaluation) is a long-running series of international workshops in natural language processing (NLP) that serve as a central venue for benchmarking semantic analysis systems. Each annual edition consists of multiple shared tasks, where research teams develop, submit, and compare computational models addressing specific semantic challenges.[1] These shared tasks have played a crucial role in shaping research directions across subfields such as sentiment analysis, semantic parsing, multilingual understanding, and text generation. In SemEval 2025, **Task 2** places a dedicated focus on *Entity-Aware Machine Translation (EA-MT)*, a problem setting that emphasizes the accurate translation of named entities such as people, locations, organizations, products, and specialized domain terms within natural language text. This task highlights the difficulty that modern translation systems often face when handling entities that require domain knowledge, cross-lingual disambiguation, or context-sensitive rendering.

To support this task, the organizers released two main datasets designed to capture a wide variety of entity-rich scenarios. The **training dataset** is derived from the Mintaka dataset [39], a large-scale, multilingual question-answering corpus notable for its diverse set of fine-grained entity types and naturally occurring language patterns. In addition, the **public and private test sets** are constructed using Wikidata,[2] a collaboratively curated international knowledge graph that provides rich structured information about entities [40, 41]. Together, these datasets ensure that systems are evaluated not only on general linguistic competence but also on their ability to preserve entity meaning, align mentions correctly, and maintain terminological consistency across languages.

Each dataset instance is formatted as a detailed JSON record and includes several key components: a unique identifier, the associated Wikidata entity ID, and a list of entity types that describe the semantic categories of the mentioned entities. The record also contains the source text, along with one or more reference target translations. Each target entry provides both the translated string and the specific entity mention

[1] `https://semeval.github.io/`

[2] `https://www.wikidata.org/`

within the translation, enabling fine-grained evaluation of entity correctness. Finally, metadata describing the source and target locales allows participants to build models sensitive to cross-lingual and cross-regional variation. An example training entry is shown below:


```
"id": "Q746666_0",
"wikidata_id": "Q746666",
"entity_types": [
  "Musical work"
],
"source": "Can you sing the chorus of the folk
song Ring a Ring o' Roses?",
"targets": [
  {
    "translation": "Puoi cantare il ritornello della canzone
    popolare Girotondo?",
    "mention": "Girotondo"
  },
  {
    "translation": "Sai cantare il ritornello del girotondo,
    la canzone popolare?",
    "mention": "girotondo"
  }
],
"source_locale": "en",
"target_locale": "it"
```


## 3.3 Methodology

### 3.3.1 Data preprocessing

Since each example already includes the entity mentions in the target language, our next step was to identify and align their counterparts in the source text. To achieve this, we employed a two-stage procedure. First, we leveraged the Qwen/Qwen2.5-VL-72B-Instruct model [42], using a quantized version to reduce computational overhead, and prompted it to predict source–target entity correspondences. Because large language models can occasionally introduce hallucinated spans, we filtered its predictions by keeping only those entity mentions that were verifiably present in the source sequence.

In parallel, we applied the AWESOME alignment framework [43], which provides token-level correspondence between source and target sentences through unsupervised cross-lingual mapping. This second method offers finer-grained structural alignment, complementing the model-based predictions from Qwen. Finally, we merged the outputs from both approaches to form a robust set of source–target entity alignments, which served as the foundation for constructing the data used in our fine-tuning stage.

### 3.3.2 Multi-task Learning fine-tuning

T5 (Text-to-Text Transfer Transformer) [44] and its multilingual extension mT5 [45] adopt a unified encoder–decoder Transformer architecture trained under a span-corruption (masked-span prediction) objective. In these models, every NLP task—whether classification, translation, question answering, or summarization—is reformulated as a text-to-text problem. This unified formulation enables a flexible multitask paradigm in which

prompts or task-specific prefixes inform the model of the desired operation, while the shared sequence-to-sequence decoder produces the corresponding output. The extensive pre-training on massive corpora, namely C4 for T5 and mC4 for mT5, equips the models with strong semantic and syntactic representations that generalize well across diverse downstream tasks. In machine translation settings, both T5 and mT5 benefit from their generative architecture: they can condition on long-range context, generalize across languages, and maintain fluency even in low-resource directions. Beyond translation, they have demonstrated competitive performance in multilingual summarization, cross-lingual question answering, entity-rich text generation, and a wide range of classification tasks.

Leveraging these strengths, our approach integrates explicit entity information into the model's input to better guide its attention toward named entities and to strengthen the connection between entity mentions and their accurate translations. During fine-tuning, we highlight entity spans detected in earlier preprocessing steps by surrounding them with lightweight XML-style boundary markers. These tags are intentionally minimal so as not to distort the source sentence or inject undesirable noise into the encoder representations. Additionally, we design a dedicated prefix for mT5 that clearly specifies both the task and the required output structure, ensuring the model follows a consistent format during supervised training.

For this task, the model is trained to generate an output sequence composed of three ordered components, separated by the "<SEP>" token: (1) the results of the NER process applied to the input text, (2) the corresponding translations of each identified entity, and (3) the full translation of the source sequence, where entity spans are again marked using the tags "`<entity>`" and "`</entity>`" to reinforce their importance. When multiple entities are present, they are concatenated using the "`|`" token to ensure a clear and consistent representation without unnecessarily increasing sequence length.

Structuring the output in this way encourages the model to first recognize and translate named entities before producing the full sentence translation, effectively foregrounding entity-related information in the decoding process. After generation, we remove the task-specific prefixes, NER outputs, entity translations, and boundary tags, leaving only the clean final translation. An example of the fine-tuning data format is shown below:

```
Input:
ner and translation: Who was the overall Commander of Allied
Forces in Europe?
Desired Output:
Europe | Allied Forces <SEP> Europa | alliierten Streitkräfte <SEP> Wer
war der Oberbefehlshaber der <entity> alliierten Streitkräfte </entity>
in <entity> Europa </entity>?
```

## 3.4 Experiments setting and results

### 3.4.1 Experimental Settings

During the evaluation phase, our experiments are conducted using the official dataset released for SemEval 2025 Task 2. Model performance is assessed primarily through the BLEU metric [25], which measures the correspondence between system outputs and human references based on n-gram similarity. To ensure a balanced and reproducible

experimental setup, the dataset is divided into training, validation, and testing subsets using fixed ratios across all participating languages, as summarized in Table 3.1.

For the fine-tuning stage, we employ the `mT5-large` model [45], consisting of approximately 580 million parameters and offering a strong multilingual baseline for sequence-to-sequence tasks. The model is trained for 50 epochs with an initial learning rate of $5 \times 10^{-5}$ and a batch size of 16, hyperparameters selected for stable optimization and efficient convergence. All experiments are executed on a single NVIDIA A40 GPU, providing sufficient computational capacity for large-scale multilingual fine-tuning.

| **Language** | **Train Size** | **Dev Size** | **Test Size** |
|---|---|---|---|
| English-German | 2615 | 654 | 818 |
| English-French | 3539 | 885 | 1107 |
| English-Italian | 2392 | 599 | 748 |
| English-Spanish | 3302 | 826 | 1032 |

Table 3.1: Dataset statistics for each language pair.

### 3.4.2 Results

The quantitative results of our experiments are summarized in Table 3.3. The table reports the translation performance of three systems—mBART, our baseline model, and the proposed multi-task learning architecture—evaluated on four language pairs: English–German, English–French, English–Italian, and English–Spanish. The scores, measured using standard translation quality metrics such as BLEU, reveal several notable trends.

Across most language directions (en–de, en–it, and en–es), the multi-task learning model delivers the strongest performance, demonstrating its ability to leverage auxiliary signals to improve entity translation and overall output quality. In contrast, for the English–French pair, the baseline system attains a slightly higher score, suggesting that the benefits of the multi-task framework may vary depending on the linguistic characteristics or data distribution of each pair. mBART, while serving as a robust pretrained multilingual model, consistently underperforms relative to the other two systems in this setting, indicating that it may require additional adaptation or more specialized fine-tuning strategies to reach competitive accuracy on this task.

| | **mBART** | **mT5 large** | **Multi-task learning** |
|---|---|---|---|
| **English-German** | 40.79 | 46.01 | **47.69** |
| **English-French** | 42.01 | **49.61** | 48.51 |
| **English-Italian** | 35.40 | 43.49 | **48.83** |
| **English-Spanish** | 45.11 | 51.12 | **54.18** |

Table 3.2: Comparison of translation performance using the BLEU score among mBART, baseline, and multi-task learning.

## 3.5 Summary of the Chapter

In this study, we presented a multi-task learning framework for Entity-Aware Machine Translation that incorporates named entity recognition (NER) signals directly into an

| | mBART | mT5 large | Multi-task learning |
|---|---|---|---|
| **English-German** | 54.94 | 55.28 | **57.51** |
| **English-French** | 55.33 | **60.39** | 59.18 |
| **English-Italian** | 50.55 | 58.02 | **59.22** |
| **English-Spanish** | 50.41 | 56.65 | **58.57** |

Table 3.3: Comparison of translation performance using M-ETA score among mBART, baseline, and multi-task learning.

mT5-based translation pipeline. By explicitly marking entity spans and generating their translations alongside the full sentence, our method encourages the model to treat entities as central components of the translation process rather than peripheral elements. This design proves particularly beneficial for handling rare, ambiguous, or domain-specific terms that conventional models often mistranslate.

Our experiments on the SemEval 2025 Task 2 benchmark demonstrate that jointly optimizing NER and machine translation within a unified architecture leads to consistent improvements over both mBART and a standard mT5 baseline. While the model exhibits slightly lower performance for certain directions, most notably English–French, the overall results highlight the effectiveness of embedding structured entity information into the translation workflow.

Looking ahead, several avenues remain open for further enhancement. These include extending the framework to cover a broader set of languages and entity categories, investigating more refined alignment or tagging strategies, and evaluating the impact of scaling to larger pretrained backbone models. Such directions hold promise for achieving even more reliable entity handling in multilingual machine translation systems.

# Chapter 4

# Entity-Aware Dataset Construction

Developing translation systems for the medical domain requires access to bilingual data in which medical terms are marked with precision. Unfortunately, such resources are extremely limited. Without parallel corpora that clearly identify diseases, procedures, medications, and other clinical concepts, it becomes difficult for models to learn how these entities should be translated. The problem extends to evaluation as well: even if a model outputs a translated medical entity, determining whether it is correct is nearly impossible without a reference set that explicitly labels these entities. To fill this gap, we created a detailed set of annotation principles designed specifically for medical terminology in English–Vietnamese parallel data. Using these guidelines, we expanded the MedEV dataset [46] into a version that includes consistent, span-level annotations, making it suitable for evaluating entity-aware translation systems.

At the same time, relying solely on manually annotated data would still leave the training process under-resourced. To supplement the gold annotations, we developed an automatic labeling process that applies the same annotation rules to the larger training corpus. This automated method produces a silver-standard dataset that mirrors the structure and constraints of the manually annotated test set. Together, the human-crafted annotation framework and the scalable automatic pipeline address the lack of annotated medical data and provide a coherent training–evaluation setup. This combined strategy offers a practical path toward building translation systems that better understand and accurately translate medical entities.

## 4.1 Annotation of Test Dataset

The MedEV dataset [46] is one of the few publicly available English–Vietnamese resources focused explicitly on the medical domain. It comprises bilingual sentence pairs drawn from medical documents, patient-facing health guidance, and clinically oriented sources (e.g., guideline summaries, informational leaflets). The corpus is partitioned into three subsets to support model development and evaluation at scale: a training set with 340,897 sentence pairs, a development (evaluation) set with 8,939 sentence pairs, and a test set with 8,960 sentence pairs. This breadth and domain focus make MedEV a natural benchmark for Vietnamese medical MT.

**Motivation for entity enrichment.** Despite its coverage, MedEV was not originally designed for *entity-aware* translation. Key notions such as diseases, anatomical structures, diagnostic and therapeutic procedures, and drug names are not annotated

at the span level, which limits direct assessment of whether systems faithfully translate clinically meaningful terminology. In high-stakes settings, global lexical metrics alone cannot reveal whether a model preserves entities (e.g., "hepatitis B", "laparoscopic cholecystectomy", "phenytoin"). To fill this gap, we construct an *enriched* version of the test split with explicit, bilingual span annotations for medical entities.

**Design principles.** Our enrichment follows three principles: (i) *clinical relevance*: annotate concepts that influence diagnosis, risk assessment, or treatment; (ii) *bilingual parity*: every English entity is aligned with a Vietnamese counterpart, allowing 1–1, 1–many, or many–1 mappings when required by linguistic differences; and (iii) *consistency*: identical inputs are annotated consistently under a documented inclusion policy (e.g., stage, dose, laterality). These principles support reproducible annotation and improve the usefulness of the dataset for evaluation and downstream error analysis.

**Stage 1: Reference lexicon consolidation.** We first build a bilingual reference lexicon that serves as the backbone for annotation. Terms are consolidated from authoritative sources:

- 215 terms from *MedlinePlus* (U.S. National Library of Medicine), covering consumer-friendly disease and procedure vocabulary[47];
- 612 entries from the *Multilingual Cancer Glossary* (Peter MacCallum Cancer Centre), covering oncology entities and synonyms[48];
- 44000 standardized Vietnamese translations of diseases, medications, and procedures from Vietnamese Ministry of Health publications [49, 50, 51, 52];

Each entry is normalized into canonical EN/VI surface forms, with optional aliases (e.g., brand vs. generic drug names). We also record `source_of_truth` (system and version) to make releases auditable.

**Stage 2: Candidate extraction via exact and near-exact matching.** We scan the EN–VI test sentences for lexicon attestations using exact and near-exact matching on both sides. Preprocessing unifies Unicode to NFC, regularizes common punctuation and spacing, and preserves diacritics in Vietnamese. Near-exact matching allows limited variation (e.g., hyphenation, plural/singular), and captures candidate spans with precise byte offsets $[start, end)$. For multi-word terms and compounds, we record the longest match and then apply policy rules to decide whether to retain a parent span or decompose into clinically meaningful subspans.

**Stage 3: Validation and alignment.** Candidate pairs are retained only if the English span and the Vietnamese span are both supported by the consolidated lexicon or an authoritative source. Ambiguous strings (e.g., general-language senses of "mass" or "shock") are discarded unless disambiguated by context or section headers. For each retained entity, we attach: (i) a type label (Disease, Anatomy, Procedure, Drug, LabTest, etc.) and (ii) an alignment label among {`1-1`, `1-many`, `many-1`, `unaligned`}. When multiple Vietnamese renderings are acceptable, we select a *canonical* VI form based on references and keep alternates as aliases to support recall without fragmenting the annotation.

**Policy highlights for consistency.** We adopt a *no-overlap* rule to prevent double-counting: compound mentions (e.g., "stage II colon cancer") are decomposed into the smallest clinically meaningful units only when component-level evidence exists; otherwise the full phrase is annotated as a single span. Abbreviations are annotated if (a) an in-sentence expansion is present or (b) there is a high-confidence lexicon mapping; cross-lingual alignment on abbreviations is performed only when the surface form is identical (e.g., "HPV" $\leftrightarrow$ "HPV"), with language-specific short forms stored as aliases. For drug names, entities are labeled with the generic name as the canonical form, and brand names are preserved as aliases; alignment is established on the generic concept. These choices make entity counts and alignments stable across documents and splits.

**Output format and reproducibility.** Annotations are stored per sentence pair in a JSONL format with fields for `src`/`tgt` text, entity lists with byte offsets on each side, type, alignment label, canonical forms, and provenance. A small tool suite validates offsets by round-trip slicing, checks schema conformance, and renders colorized spans for spot checks. All lexicon sources are pinned in the release manifest to support exact regeneration of the test annotations.

**Descriptive statistics.** The resulting enriched test split enables precise measurement of entity-level translation accuracy in high-risk medical contexts. Across the MedEV test set of 8,960 sentence pairs, we annotated 3,675 pairs (41.0%). Among these, 2,108 pairs (23.53% of the full set) contain at least one aligned entity. In total, we produced 5,200 source–target entity annotations. Subsequent sections report examples illustrating policy decisions (e.g., decomposition vs. retention and shared-form abbreviations).

### 4.1.1 Practical Issues in Entity Annotation

This subsection summarizes the practical issues we repeatedly encounter when annotating EN–VI medical entities and fixes the policy choices we use to keep spans and alignments stable for evaluation. First, compound mentions trigger *overlaps/nesting/double-counting*; to avoid inflated counts, we adopt a *no-overlap* rule and only *decompose* a phrase into smaller, clinically meaningful units when authoritative sources (e.g., MoH documents, Peter MacCallum glossary) provide component-level evidence; otherwise the full phrase is retained as a single entity. Second, for *abbreviations*, we annotate an acronym only when it has an in-sentence expansion or a high-confidence lexicon mapping, and treat it as an aligned entity across languages only if the surface form is identical (shared-form); language-specific short forms are stored as *aliases*, not as aligned entities. Third, where *brand and generic drug names* co-occur, we use the generic name as the canonical entity label, use brands solely as aliases, and align on the generic concept to prevent label fragmentation. Finally, to handle *synonyms and paraphrases* that escape exact matching, we combine curated, type-specific alias lists with guarded fuzzy matching; we select a single canonical Vietnamese surface (per references) for alignment and keep other acceptable renderings as aliases, escalating any case that shifts clinical scope for adjudication. The examples that follow instantiate these rules on "colorectal cancer" vs. its parts, HPV/NST abbreviation behavior, generic antiepileptic drug lists, and paraphrastic translations of "traditional herbal medicines".

**Overlaps, nesting, and double-counting.** Compound mentions such as "stage II colon cancer" invite inconsistent treatment: some annotators label the whole phrase, others label its parts (stage, organ, disease). This inflates counts and complicates training and evaluation. We therefore commit to a single, reproducible policy: *no overlapping spans*. When authoritative references provide component-level translations, we *decompose* the phrase into its smallest clinically meaningful units; when such evidence is lacking, we retain the full phrase as a single entity. This avoids double-counting while preserving clinical meaning.

*Example 1 (decompose).*

> **EN:** "Colon diseases are increasingly detected due to the development of colonoscopic techniques, especially colorectal cancer."
> **VI:** "Bệnh lý đại tràng ngày càng phát hiện được nhiều nhờ vào phương tiện nội soi đại tràng, đặc biệt ung thư đại trực tràng."

From Vietnamese Ministry of Health materials and the Peter MacCallum Vietnamese glossary, we confirm the canonical pair "colorectal cancer" ↔ "ung thư đại trực tràng", and we also have independent evidence that "cancer" ↔ "ung thư". Consequently, for finer-grained measurement we *split* the parent phrase into two non-overlapping entities: "colorectal" ↔ "đại trực tràng" and "cancer" ↔ "ung thư". The parent span is *not* additionally annotated, preventing double-counting.

*Example 2 (retain as a whole).*

> **EN:** "Complications occurred in 2 patients with neurological deficit in 1 patient (3.3), cerebral hemorrhage after embolization in 1 patient (3.3), mortality (0)."
> **VI:** "Biến chứng gặp trong 2 bệnh nhân với 1 trường hợp có thiếu hụt thần kinh và một trường xuất huyết não (3,3), không có trường hợp nào tử vong."

Here, official MoH documents support the phrase-level mapping "neurological deficit" ↔ "thiếu hụt thần kinh", but we find no reliable authority that separately codifies "neurological" or "deficit" as standalone medical entities in this context. We therefore *keep* the phrase as a single entity pair without decomposition.

**Abbreviations.** Acronyms (e.g., HBV, ALT, CT) are ubiquitous but often appear without an in-sentence expansion; their meaning can vary by section and domain. Blindly labeling such forms risks incorrect codes and noisy alignments. Our policy is twofold: (i) an abbreviation is annotated only if it has an in-sentence expansion or a high-confidence lexicon mapping; otherwise it is marked *uncertain* and queued for adjudication, and (ii) for bilingual alignment we only treat an abbreviation as the aligned entity when its surface form is *identical* in English and Vietnamese (a "shared-form" abbreviation). When forms differ across languages, we annotate the canonical term on each side and store the language-specific abbreviation as an *alias*, not as an aligned entity. An EN↔VI alias table is maintained to stabilize future matches.

*Example 1 (keep the abbreviation).*

> **EN:** "Results and conclusions: The rate of HPV positive is 12.75."
> **VI:** "Kết quả và kết luận: Tỉ lệ dương tính với HPV 12,75."

The abbreviation "HPV" (human papillomavirus) is unchanged across languages, so we annotate the entity pair as "HPV" ↔ "HPV" (canonical concept: human papillomavirus).

*Example 2 (use alias, not aligned abbreviation).*

> **EN:** "Chromosomal mosaicism in prenatal diagnosis is a complex problem ..."
> **VI:** "Thể khảm nhiễm sắc thể (NST) trong chẩn đoán trước sinh ..."

Here, the Vietnamese text introduces a language-specific abbreviation "NST" for "nhiễm sắc thể" (chromosomal). Because no corresponding abbreviation appears on the English side, we annotate the aligned entity as "chromosomal" ↔ "nhiễm sắc thể" and record "NST" as a Vietnamese alias only; we do *not* create an aligned pair "chromosomal" ↔ "NST".

**Brand vs. generic drug names.** Source texts often mix generic names and brand names (e.g., *Tylenol* vs. *paracetamol/acetaminophen*), and Vietnamese usage typically prefers a consistent canonical generic form. Without a clear policy, annotations fragment across surface variants and undermine consistency. Our rule is: *label the entity with the generic name as the canonical form*; record any brand names that appear as *aliases*, and document the preferred Vietnamese generic form so the same pharmacological concept always maps to a single canonical surface string on each side. When both brand and generic occur in the same sentence pair, alignment is established on the generic concept; brand strings are preserved only as aliases.

*Example (all generics, identical across EN–VI).*

> **EN:** "The use of aromatic antiepileptic drugs such as Carbamazepine, Oxcarbazepine, Phenytoin, Phenobarbital, Lamotrigine, Primidone and Zonisamide is often associated with skin rash and other symptoms and signs of drug hypersensitivity."
> **VI:** "Việc sử dụng các thuốc chống động kinh có vòng thơm (aromatic) như Carbamazepine, Oxcarbazepine, Phenytoin, Phenobarbital, Lamotrigine, Primidone và Zonisamide thường có liên quan tới phát ban trên da và các triệu chứng, dấu hiệu khác của quá mẫn do thuốc."

All listed items are generic drug names whose surface forms are unchanged in Vietnamese. We therefore annotate each drug as a `Drug` entity with a 1–1 alignment; brands, if mentioned elsewhere, would be stored under `aliases` but not used for alignment. Minimal JSON-style records for two items:

```
...,
{
  "source_entity": "Phenytoin",
  "target_entity": "Phenytoin",
},
{
  "source_entity": "Phenobarbital",
  "target_entity": "Phenobarbital",
},
....
```

If a brand surface (e.g., *Tegretol* for carbamazepine) appeared on one side, we would still align on the generic *carbamazepine* and record the brand only as an alias of that concept.

**Synonyms and paraphrases missed by exact match.** Clinically correct entities are frequently realized as paraphrases or near-synonyms (e.g., "flu" vs. "influenza") or by vernacular Vietnamese. Pure exact/near-exact matching under-annotates these cases, depressing recall and biasing evaluation toward lexiconized forms. Our policy pairs a curated, type-specific alias list with *guarded* fuzzy matching (character/word similarity plus whitelists and stoplists). When multiple Vietnamese renderings are acceptable, we (i) select a *canonical* VI surface form based on authoritative references, (ii) annotate alignment on the canonical pair, and (iii) retain alternates as *aliases* to support detection without fragmenting the annotation. If a paraphrase shifts clinical scope or specificity, the case is escalated for adjudication rather than auto-merged.

*Example (canonicalize, keep alias).*

> **EN:** "In order to contribute to the detection of banned substances in *traditional herbal medicines*, we conducted a study on the method of quantification of diclofenac sodium in *traditional herbal medicines* ..."
> **VI:** "Để góp phần vào công tác kiểm tra phát hiện các chất cấm trộn lẫn trong *thuốc đông dược*, chúng tôi tiến hành đề tài xây dựng phương pháp định lượng diclofenac natri lẫn trong *chế phẩm đông dược* ..."

The English term "traditional herbal medicines" appears twice; the Vietnamese renders it once as "thuốc đông dược" and once as "chế phẩm đông dược". Both are intelligible in context, but our reference sources prefer "thuốc đông dược" as the canonical form. We therefore annotate the aligned entity as

$$\text{"traditional herbal medicines"} \leftrightarrow \text{"thuốc đông dược"},$$

and record "chế phẩm đông dược" as a Vietnamese *alias*. This preserves recall (via alias-aware matching) while preventing label fragmentation in evaluation.

## 4.2 Silver-labeled Dataset Construction

Although the manually annotated test set provides a dependable and carefully validated resource for measuring medical entity translation quality, it is insufficient for training a robust translation model. The annotated portion covers only a small fraction of the medical terminology that appears in real-world texts, meaning that the model cannot learn the full diversity of diseases, medications, anatomical references, and clinical procedures it may encounter. Extending the same manual annotation workflow to the entire MedEV training split is also impractical: the process requires cross-referencing multiple medical glossaries, verifying bilingual consistency, and marking precise span boundaries, all of which demand considerable expert labor and computational resources. To address these constraints, we develop an automated pipeline for constructing a **silver-standard training dataset**. This dataset mirrors the annotation style and criteria of the manually curated test set while enabling annotation at scale. The pipeline consists of four main stages, described in detail below.

- **NER extraction.** The pipeline begins with identifying medically relevant terms in the English portion of the bilingual corpus. We apply a pretrained medical-domain named entity recognition (NER) model capable of detecting a variety of clinical entity types, including diseases, symptoms, anatomical structures, diagnostic and therapeutic procedures, pharmaceutical compounds, and pathogens. The model

produces token-level spans that serve as initial candidates for alignment. When Vietnamese reference translations are available, we also apply NER on the target side to capture additional entity cues and strengthen the cross-lingual extraction process.

- **LLM-based alignment.** After extracting source-language entity spans, we use a large language model (LLM) to propose initial source-target entity mappings. The LLM is prompted with the English entity spans and instructed to generate their Vietnamese equivalents in a controlled, structured format. To limit hallucinations and overly creative interpretations, we enforce strict prompting rules requiring that outputs remain medically valid, follow a prescribed output structure, and avoid paraphrasing the original entity. This step leverages the LLM's multilingual and biomedical knowledge to provide a preliminary alignment signal. The prompt we use to extracted is as following: The prompt we used to extract and aligne the entities is as:

```
    """
    You are a helpful assistant for medical entity alignment between parallel
↪   sentences. I have a sentence in a source language and its corresponding
↪   sentence in a target language.

    Source Sentence: "@@@@src_sentence@@@@"
    Target Sentence: "@@@@tgt_sentence@@@@"

    There is an entity in the source sentence: "@@@@entity@@@@".

    Task:
    1. Identify all medically relevant entities in the source sentence.
    2. For each source entity, locate and extract the exact corresponding
↪   token span in the target sentence.
    3. Produce only a JSON-encoded list of objects, each with the following
↪   keys:
       - "source_entity": the entity string as it appears in the source
    ↪   sentence
       - "target_entity": the matching entity string in the target sentence

    Guidelines:
    - Keep only medically relevant entities, such as:
      - Syndromes, organs, diseases, chemical substances, drugs, treatment
  ↪   methods, or medical metrics.
      - Non-human living organisms (e.g., bacteria, viruses, laboratory
  ↪   animals).
      - Medical departments or specialties.
    - If a source entity contains multiple alternatives separated by a slash
↪   ("/") or space-slash-space ("A/B" or "A / B"), treat each alternative as a
↪   separate entity. Create one JSON object per split entity, matching each
↪   split source token to the corresponding split target token. Example:
      For Example:
      1. Input pair: {"source_entity": "CE / CLM syndrome", "target_entity":
  ↪   "Hội chứng CE / CLM"}
        -> Output two objects: {"source_entity": "CE", "target_entity": "CE"}
    ↪   and {"source_entity": "CLM", "target_entity": "CLM"}.
      2. Input pair: {"source_entity": "B-cell/T-cell
  ↪   lymphoma","target_entity": "lymphoma tế bào B / tế bào T"}
```

```
        ->  Output two objects: {"source_entity": "B-cell lymphoma",
    ↪  "target_entity": "lymphoma tế bào B"} and {"source_entity": "T-cell
    ↪  lymphoma", "target_entity": "lymphoma tế bào T"}
      3. Input pair: {  "original_source_entity": "PNE/SEV",
  ↪  "original_target_entity": "PNE/SEV",
         -> Output two objects: {"source_entity": "PNE", "target_entity":
     ↪  "PNE"} and {"source_entity": "SEV", "target_entity": "SEV"}
      - If the target uses the same slash-separated structure, split the
  ↪  target the same way and align by position.
    - **Do not extract or include anything that appears inside parentheses.**
      - For example, in "Magnetic resonance imaging (MRI)", do **not**
  ↪  consider "MRI" as an entity.
      - Always skip abbreviations, acronyms, or additional notes in
  ↪  parentheses.
    - Remove entities that are:
      - Clearly unrelated to the medical domain (e.g., locations, common
  ↪  people, date, time metric or lifestyle behaviors not mentioned in a
  ↪  medical context).
      - Pure numbers, percentages, or amounts.
      - Not semantically similar in meaning between the source and target
  ↪  sentences.
    - If no valid or relevant entity pair is found, return an empty list [].
    - Do not include any text, explanations, or comments outside the JSON
↪  output.

    Examples - High priority (should be kept):
    [
      {"source_entity": "endoscopic", "target_entity": "hình ảnh nội soi"},
      {"source_entity": "urlological features", "target_entity": "mô bệnh
  ↪  học"},
      {"source_entity": "upper gastrointestinal", "target_entity": "thực
  ↪  quản"}
    ]

    Should be removed (ignore):
    [
      {"source_entity": "21, 4%", "target_entity": "21,4%"},
      {"source_entity": "eat snack in the evening", "target_entity": "ăn thêm
  ↪  bữa phụ vào buổi tối"},
      {"source_entity": "Total energy intake", "target_entity": "Tổng năng
  ↪  lượng"},
      {"source_entity": "hospital stay", "target_entity": "thời gian nằm
  ↪  viện"},
      {"source_entity": "Hanoi medical university hospital", "target_entity":
  ↪  "Bệnh viện Đại học Y Hà Nội"},
      {"source_entity": "males", "target_entity": "trai"}
      {"source_entity": "7/2011","target_entity": "tháng 7/2011"},
      {"source_entity": "ng / ml","target_entity": "ng / ml"}
      {"source_entity": "patients","target_entity": "trẻ nhỏ"}
      {"source_entity": "elderly","target_entity": "người cao tuổi (NCT)"}
      {"source_entity": "Hb < 10 g / dl","target_entity": "Hb < 10 g / dl"}
    ]

    Expected output format (JSON only):
    [
      {"source_entity": "hypertension", "target_entity": "tăng huyết áp"},
      {"source_entity": "beta-blockers", "target_entity": "thuốc chẹn -giao
  ↪  cảm"}
```

```
    ]
    """
```

- **Sentence-pair mining.** To complement the LLM alignment results, we instruct the LLM to extract (source entity, target entity) pairs directly from full English–Vietnamese sentence pairs *without providing explicit hints*. This forces the model to ground its predictions in actual sentence context rather than relying solely on dictionary-like mappings. We constrain the extraction to medically relevant categories included in our validated lexicon, such as diseases, organs, medical procedures, parasites, medications, drugs, and chemical substances, to reduce noise. Furthermore, we supply a small set of few-shot examples sourced from authoritative medical glossaries to guide the LLM toward producing accurate and category-consistent extractions. The prompt I used is as following:

```
    """
    You are a specialized assistant for biomedical entity alignment in
↪   parallel sentences.

    Input:
    Source Sentence: "@@@@src_sentence@@@@"
    Target Sentence: "@@@@tgt_sentence@@@@"

    Task:
    1. Identify all medically relevant entities in the source sentence.
    2. For each source entity, locate and extract the exact corresponding
↪   token span in the target sentence.
    3. Produce only a JSON-encoded list of objects, each with the following
↪   keys:
       - "source_entity": the entity string as it appears in the source
    ↪  sentence
       - "target_entity": the matching entity string in the target sentence

    Guidelines:
    - Keep only medically relevant entities, such as:
      - Syndromes, organs, diseases, chemical substances, drugs, treatment
  ↪   methods, or medical metrics.
      - Non-human living organisms (e.g., bacteria, viruses, laboratory
  ↪   animals).
      - Medical departments or specialties.
    - If a source entity contains multiple alternatives separated by a slash
↪   ("/") or space-slash-space ("A/B" or "A / B"), treat each alternative as a
↪   separate entity. Create one JSON object per split entity, matching each
↪   split source token to the corresponding split target token. Example:
      For Example:
      1. Input pair: {"source_entity": "CE / CLM syndrome", "target_entity":
  ↪   "Hội chứng CE / CLM"}
        -> Output two objects: {"source_entity": "CE", "target_entity": "CE"}
    ↪   and {"source_entity": "CLM", "target_entity": "CLM"}.
      2. Input pair: {"source_entity": "B-cell/T-cell
  ↪   lymphoma","target_entity": "lymphoma tế bào B / tế bào T"}
        ->  Output two objects: {"source_entity": "B-cell lymphoma",
    ↪   "target_entity": "lymphoma tế bào B"} and {"source_entity": "T-cell
    ↪   lymphoma", "target_entity": "lymphoma tế bào T"}
```

```
    3. Input pair: {  "original_source_entity": "PNE/SEV",
↪  "original_target_entity": "PNE/SEV",
       -> Output two objects: {"source_entity": "PNE", "target_entity":
    ↪  "PNE"} and {"source_entity": "SEV", "target_entity": "SEV"}
    - If the target uses the same slash-separated structure, split the
↪  target the same way and align by position.
  - **Do not extract or include anything that appears inside parentheses.**
    - For example, in "Magnetic resonance imaging (MRI)", do **not**
↪  consider "MRI" as an entity.
    - Always skip abbreviations, acronyms, or additional notes in
↪  parentheses.
  - Remove entities that are:
    - Clearly unrelated to the medical domain (e.g., locations, common
↪  people, date, time metric or lifestyle behaviors not mentioned in a
↪  medical context).
    - Pure numbers, percentages, or amounts.
    - Not semantically similar in meaning between the source and target
↪  sentences.
  - If no valid or relevant entity pair is found, return an empty list [].
  - Do not include any text, explanations, or comments outside the JSON
↪  output.

  Examples - High priority (should be kept):
  [
    {"source_entity": "endoscopic", "target_entity": "hình ảnh nội soi"},
    {"source_entity": "urlological features", "target_entity": "mô bệnh
↪  học"},
    {"source_entity": "upper gastrointestinal", "target_entity": "thực
↪  quản"}
  ]

  Should be removed (ignore):
  [
    {"source_entity": "21, 4%", "target_entity": "21,4%"},
    {"source_entity": "eat snack in the evening", "target_entity": "ăn thêm
↪  bữa phụ vào buổi tối"},
    {"source_entity": "Total energy intake", "target_entity": "Tổng năng
↪  lượng"},
    {"source_entity": "hospital stay", "target_entity": "thời gian nằm
↪  viện"},
    {"source_entity": "Hanoi medical university hospital", "target_entity":
↪  "Bệnh viện Đại học Y Hà Nội"},
    {"source_entity": "males", "target_entity": "trai"}
    {"source_entity": "7/2011","target_entity": "tháng 7/2011"},
    {"source_entity": "ng / ml","target_entity": "ng / ml"}
    {"source_entity": "patients","target_entity": "trẻ nhỏ"}
    {"source_entity": "elderly","target_entity": "người cao tuổi (NCT)"}
    {"source_entity": "Hb < 10 g / dl","target_entity": "Hb < 10 g / dl"}
  ]

  Expected output format (JSON only):
  [
    {"source_entity": "hypertension", "target_entity": "tăng huyết áp"},
    {"source_entity": "beta-blockers", "target_entity": "thuốc chẹn -giao
↪  cảm"}
  ]
  """
```

- **Merge & filter.** In the final stage, we consolidate the outputs from both the LLM alignment and sentence-pair mining steps. Entity pairs that appear in both methods, match entries in the curated bilingual lexicon, or exhibit strong lexical and semantic coherence are retained. Pairs showing inconsistent meanings, structural mismatches, or medically implausible translations are discarded. The result is a high-quality **silver-standard training dataset** that significantly expands the quantity of annotated examples while preserving consistency with the annotation principles used for the manually labeled test set.

Through this multi-stage pipeline, we are able to generate large-scale, medically grounded entity annotations suitable for training domain-aware translation models without requiring extensive human effort.

### 4.2.1 Evaluation of silver data

To assess the quality of entity annotations in the silver dataset, we conducted a manual evaluation on a randomly sampled subset of the validation set. Specifically, we selected 200 sentence pairs that contained at least one automatically generated entity annotation and manually verified the correctness of each annotated entity according to the annotation guidelines and the medical terminology reference documents.

The objective of this evaluation was to estimate the precision of the silver annotations, which measures the proportion of generated entity annotations that are correct. An annotation was considered correct if both the source-language entity and its corresponding target-language entity accurately represented the same medical concept and their boundaries followed the annotation guidelines.

The evaluation sample contained a total of 646 annotated entities. Among them, 227 annotations were identified as incorrect, resulting in 419 correct annotations. Therefore, the precision of the silver dataset was calculated as follows:

$$\text{Precision} = \frac{\text{Correct Annotations}}{\text{Total Annotations}} = \frac{419}{646} = 64.86$$

To better understand the limitations of the annotation pipeline, we further analyzed all incorrect annotations and categorized them according to their underlying causes. The distribution of error types is presented in Table 4.1.

Table 4.1: Error analysis of the silver dataset sample

| **Error Type** | **Count** | **Percentage** |
|---|---|---|
| Boundary errors | 105 | 46.3% |
| Generic terms | 60 | 26.4% |
| Abbreviation errors | 31 | 13.7% |
| Procedural action nominalization | 25 | 11.0% |
| Prognostic statements | 6 | 2.6% |
| **Total** | **227** | **100.0%** |

As shown in Table 4.1, boundary errors account for nearly half of all incorrect annotations (46.3

Based on the analysis, five major categories of annotation errors were identified: boundary errors, generic terms, abbreviation errors, procedural action nominalizations, and prognostic statements. The definitions and representative examples of each error type are described below.

- **Boundary Errors**

  Boundary errors occur when the annotation span includes additional tokens that do not contribute to the core medical concept or when the annotated span does not conform to the annotation guidelines. Such errors typically arise when modifiers, severity indicators, temporal expressions, or contextual descriptors are incorrectly included as part of the entity.

  For example, in the phrase *"severe renal dysfunction"*, the annotation system labeled the entire phrase as an entity. However, according to the annotation guidelines, the core medical concept is *"renal dysfunction"*, while *"severe"* only expresses the severity level of the condition and should not be included within the entity boundary.

  > Annotated entity: **severe renal dysfunction** Preferred entity: renal dysfunction

- **Generic Terms**

  This error occurs when the system annotates generic objects, instruments, or general terms instead of medically specific concepts. Although these terms may appear in medical documents, they do not represent specialized medical terminology and therefore provide limited value for medical terminology extraction.

  For example, the word *"cane"* was annotated as an entity:

  > English: cane Vietnamese: gậy

  While the term appears in a rehabilitation context, it refers to a common assistive device rather than a specialized medical concept. Similar cases were classified as generic term errors.

- **Procedural Action Nominalization**

  This category refers to cases where the annotation corresponds to a generalized medical action or procedure name rather than a specific medical procedure. These terms are often overly broad and become semantically similar to generic vocabulary after translation.

  For example:

  > English: surgery Vietnamese: phẫu thuật

  The term *"surgery"* denotes a general treatment action rather than a specific surgical procedure. In contrast, procedure names such as *"thyroidectomy"* or *"laparoscopic nephrectomy"* would represent valid medical entities because they identify specific interventions.

- **Abbreviation Errors**

  Abbreviation errors occur when an abbreviation in one language corresponds to its expanded form in the other language. Although the translation is semantically correct, the resulting annotation pair does not represent a direct terminology correspondence. Such cases may introduce noise into bilingual terminology extraction because the abbreviated form alone often lacks sufficient lexical information.

  For example:

English: LUTS  Vietnamese: triệu chứng đường tiểu dưới

Here, *LUTS* is an abbreviation for *Lower Urinary Tract Symptoms*. The English side contains an abbreviated expression while the Vietnamese side contains the expanded meaning. To improve annotation consistency and reduce ambiguity during model training, such cases are currently excluded from the silver dataset.

- **Prognostic Statements**

  Prognostic statement errors occur when the annotation corresponds to a clinical outcome, prognosis, or evaluative statement rather than a medical entity. These expressions describe the expected progression, severity, or outcome of a disease rather than a concrete medical concept.

  For example:

  English: high mortality  Vietnamese: tỷ lệ tử vong cao

  The phrase expresses a prognostic assessment of patient outcomes rather than a disease, symptom, treatment, anatomical structure, or medical procedure. Consequently, it falls outside the scope of the target entity categories and is treated as an annotation error.

Overall, the error analysis reveals that the majority of annotation errors originate from boundary detection issues and insufficient filtering of non-terminological expressions. These findings suggest that future improvements should focus on refining entity boundary identification, enhancing terminology specificity filtering, and developing more robust handling strategies for abbreviations and generalized procedural expressions.

## 4.3 Limitations

This section focuses only on limitations that follow directly from our pipeline described earlier: (i) enriching the MedEV test split with span-level entities using a bilingual lexicon consolidated from *MedlinePlus* (215 terms), the *Multilingual Cancer Glossary* (612 entries), and the Vietnamese Ministry of Health list (44,000 standardized items), and (ii) constructing a silver-standard training set via **NER** extraction, **LLM**-based alignment and sentence-pair mining, followed by **merge & filter**.

### 4.3.1 Limits in Test-Set Annotation

**Coverage bias.** Our reference backbone is necessarily incomplete: the 215/612/44,000 items cover high-salience concepts but under-represent rare diseases, pediatric terms, device brands, legacy Vietnamese spellings, and evolving oncology nomenclature. Exact or near-exact matching will therefore *miss* valid mentions that appear as paraphrases, inflections, or uncommon synonyms.

**Boundary sensitivity.** Despite NFC normalization and explicit rules, span boundaries can drift in edge cases: dose attachment to drugs ("metformin 500 mg"), stage/grade inclusion ("stage II"), hyphenated or slashed forms ("IgG/IgM"), and parenthetical aliases ("hepatitis B (HBV)"). Small boundary choices propagate to span-level metrics and may change entity-level accuracy by a few points for affected categories.

**Cross-lingual alignment strain.** English↔Vietnamese reordering (e.g., causal "X-induced Y" ↔ "Y do X"), one-to-many decompositions on the Vietnamese side, and generic↔specific mappings ("acetaminophen" ↔ "paracetamol") can yield legitimate translations that our strict bilingual lexicon does not list verbatim. This risks false negatives even when the medical meaning is correct.

### 4.3.2 Limits in Silver-Set Construction

**NER extraction noise.** Out-of-domain or language-mismatched NER models (especially on Vietnamese) produce false positives (generic nouns tagged as entities) and false negatives (multiword or discontinuous medical names). Category sets may not align perfectly with our taxonomy (e.g., "clinical finding" vs. Disease/Symptom split).

**LLM alignment & mining.** Even under constrained prompts, LLMs may hallucinate canonical forms, over-normalize brand→generic names, or prefer consumer-style paraphrases. When sentence-pair mining is performed *without hints*, extraction quality depends on discourse context and can degrade for long sentences or heavily reordered pairs.

**Merge & filter trade-offs.** Agreement heuristics (LLM-alignment ∩ sentence-mined pair, plus semantic similarity and lexicon support) reduce but do not eliminate noise. Thresholds that are too strict discard useful long-tail entities; thresholds that are too lax admit subtle mistranslations (e.g., organism vs. disease name). Confidence is also affected by EN:VI length ratios and reordering, which can depress overlap-based signals.

## 4.4 Summary of the Chapter

This chapter addressed the central data bottleneck for building and evaluating entity-aware English–Vietnamese (EN–VI) medical translation systems. We motivated the need for span-level supervision of clinically salient concepts (diseases, anatomical structures, procedures, drugs) and introduced an enriched version of the MedEV test split [46] with explicit, bilingual entity annotations. The enrichment is grounded in authoritative resources: *MedlinePlus* (215 items), the *Multilingual Cancer Glossary* (612 entries), and Vietnamese Ministry of Health publications (44,000 standardized terms), yielding a reference lexicon that anchors both matching and adjudication. The resulting test annotations are stored with byte-based half-open offsets, canonical EN/VI surface forms, and alignment labels (1–1/1–many/many–1), ensuring reproducibility and auditability.

We formalized a set of consistency policies that stabilize counts and alignments across documents: a *no-overlap* rule for compound mentions with evidence-based decomposition; abbreviation handling that requires in-sentence expansion or high-confidence lexicon support and treats only shared-form acronyms as aligned entities. Practical pitfalls and their remedies (span boundaries, synonymy/paraphrase, Unicode/offset integrity, token-vs-character indices, and cross-lingual reordering) were enumerated and operationalized (§4.1.1), so identical inputs yield identical spans under a documented inclusion policy.

To overcome the scarcity of gold labels for training, we designed a scalable pipeline to construct a **silver-standard** EN–VI training set that mirrors the test-set guidelines. The pipeline couples biomedical NER on both sides of the parallel text with two LLM

passes: (i) span-conditioned alignment and (ii) sentence-pair mining without hints. A merge–and–filter stage retains entity pairs supported by dual evidence or lexicon agreement and discards semantically inconsistent or medically implausible mappings. All artifacts capture provenance (NER/LLM/mining/lexicon) to support downstream confidence estimation and error analysis.

Empirically, across the MedEV test set of 8,960 sentence pairs, we annotated 3,675 pairs (41.0%); among these, 2,108 pairs (23.53% of the full set) contain at least one aligned entity. In total, the chapter produced 5,200 source–target entity annotations, providing a reliable basis for entity-level evaluation in high-risk clinical contexts. We also documented known limitations: lexicon coverage bias, residual boundary sensitivity, alignment strain from legitimate EN↔VI reorderings, NER noise in low-resource settings, and LLM over-normalization. These are mitigated, though not eliminated, via deterministic policies, alias-aware canonicalization, reordering-aware alignment labels, and strict JSON-only prompts with exclusion rules (e.g., parentheses, units, dates).

Together, the enriched test set and silver-standard training corpus establish a coherent training–evaluation substrate for the remainder of this work. Subsequent chapters build on these resources to define entity-sensitive scoring metrics, analyze model behavior at the concept level, and integrate the annotations into learning objectives for entity-forward machine translation.

# Chapter 5

# Multi-Aspect Reinforcement Learning Machine Translation

The goal of this chapter is to provide a detailed description of the Multi-aspect Reinforcement Learning method and the motivation, as well as how this method helps solve the existing problems in the Entity-Aware Neural Machine Translation method.

## 5.1 Overview of Approach

We cast entity-aware machine translation as reinforcement learning and train the policy with **Group Relative Policy Optimization (GRPO)**. As sketched in Fig. 5.1, the translation model acts as a policy $\pi_\theta(y \mid x)$ that, for each source sentence $x$, samples $K$ candidate hypotheses. Each hypothesis is scored by complementary reward modules that target distinct facets of quality: lexical fluency, entity fidelity, attention alignment, and syntactic/structural preservation. GRPO avoids dependence on absolute reward magnitudes; instead, it compares candidates *within* each group and assigns relative advantages from their within-group ranking. This ranking-based update reduces sensitivity to reward scale and eliminates the need for a KL-anchored reference policy, leading to more stable optimization.

Given $\mathcal{D} = \{(x_i, y_i^\star)\}$ and group size $K$, the policy produces $\{\hat{y}_1, \ldots, \hat{y}_K\}$ for each $x$, and each sample receives the composite reward

$$R_{\text{total}} = \alpha R_{\text{BLEU}} + \beta R_{\text{M-ETA}} + \delta R_{\text{attn}} + \eta R_{\text{dep}}. \tag{5.1}$$

GRPO then normalizes rewards within the group and computes advantages by contrasting each sample against a *group baseline* (e.g., the mean or a top-$p$ average). This yields stable gradients even when reward distributions vary widely across sentences or languages: updates push the policy toward candidates with higher relative reward and away from weaker alternatives. Because different reward components can dominate on different examples, the groupwise comparison naturally supports our multi-objective setting—encouraging the model to imitate the strongest candidate in each group.

Overall, this framework operationalizes our goal: *improve named-entity fidelity, attention consistency, and structural accuracy without sacrificing general fluency.*

## 5.2 Sentence-level Lexical Matching Reward

To preserve global fluency and prevent drift from general-domain translation quality, we include a *Lexical Matching Similarity* reward based on sentence-level BLEU score.

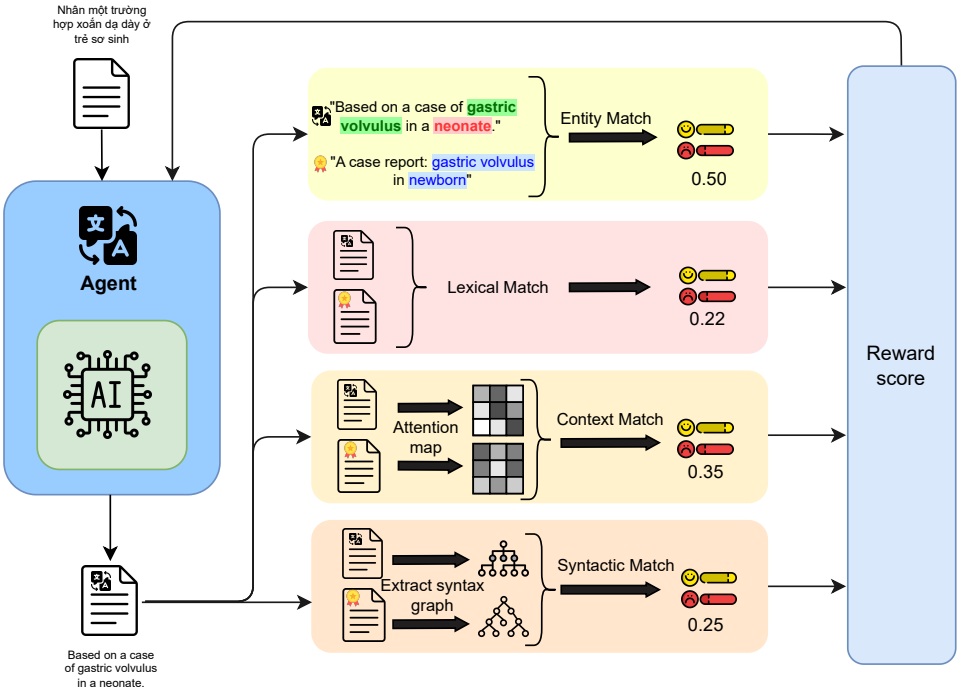


Figure 5.1: Overall architecture of the proposed reinforcement learning translation system.

For a hypothesis $\hat{y}$ and reference $y^\star$, the BLEU score summarizes the degree of $n$-gram overlap while accounting for length adequacy via a brevity penalty, so higher scores reflect closer lexical fidelity without rewarding under-translation.

$$R_{\text{BLEU}} = \frac{\text{BLEU}(\hat{y},\, y^\star)}{100} \in [0, 1]. \tag{5.2}$$

**Role in GRPO.** Within GRPO, absolute magnitudes of $R_{\text{BLEU}}$ are less important than each hypothesis's *relative ranking* within the sampled group for the same source. A hypothesis that attains a higher BLEU score than its siblings receives a positive relative advantage, nudging the policy toward outputs with stronger lexical fidelity and length adequacy; conversely, lower-ranked hypotheses receive negative advantages. In practice, this pressure helps discourage omissions (penalized by the brevity component), unsupported insertions, and overly literal distortions that deviate from the reference.

**Complementarity.** Lexical Matching serves as the backbone of the global reward, anchoring the model to fluent, reference-consistent translation patterns. It complements entity-focused and structure-focused rewards by providing a broad lexical signal, while those localized rewards target correctness around entities and relational structure.

## 5.3 Entity-level Matching Reward

To directly improve named-entity translation accuracy, we incorporate an *Entity Matching Reward* computed via the M-ETA score. Let $E_{\text{ref}} = \{e_1, \ldots, e_n\}$ be the set of gold entities associated with a source $x$ and reference translation $y^\star$. For a model hypothesis

$\hat{y}$, we count how many reference entities are correctly realized under a suite of normalization rules (case folding, canonical forms, transliteration, and diacritic tolerance). The basic recall-style reward is

$$R_{\text{M-ETA}} \;=\; \begin{cases} \frac{c}{|E_{\text{ref}}|}, & |E_{\text{ref}}| > 0, \\ 1, & |E_{\text{ref}}| = 0, \end{cases} \tag{5.3}$$

where $c$ is the number of entities in $E_{\text{ref}}$ that appear correctly in $\hat{y}$ after normalization.

**Entity extraction and canonicalization.** Entities are harvested from $x$ and $y^\star$ using a clinical/biomedical NER (or a domain NER for technical texts) and optionally aligned to establish gold target surface forms. Before matching, we apply a canonicalizer canon$(\cdot)$ that: (i) lowercases and strips punctuation; (ii) removes or normalizes diacritics; (iii) maps synonyms/aliases to a preferred label (e.g., drug brand $\rightarrow$ INN); (iv) normalizes Greek letters and units (e.g., "$\mu$g" $\leftrightarrow$ "mcg", "mg/dL" $\leftrightarrow$ "mg per dL"); and (v) applies language-appropriate transliteration when an entity must be transliterated rather than translated. A match is declared if canon$(e_i)$ is a substring of canon$(\hat{y})$ or passes a soft string-similarity threshold (e.g., token Jaccard or normalized edit distance), which grants tolerance to boundary jitter and minor orthographic variations.

**Integration with GRPO.** Within Group Relative Policy Optimization, for each source $x$ we sample a group of $M$ hypotheses $\{\hat{y}_j\}_{j=1}^{M}$. We compute per-sample entity rewards $r_j^{\text{ent}} = \lambda_{\text{ent}}\, R_{\text{M-ETA}}(x, \hat{y}_j)$ (or $R^{F_1}_{\text{M-ETA}}$), and form centered advantages

$$a_j^{\text{ent}} \;=\; r_j^{\text{ent}} \;-\; \frac{1}{M}\sum_{m=1}^{M} r_m^{\text{ent}}. \tag{5.4}$$

These advantages enter the clipped GRPO/PPO objective alongside other rewards (e.g., BLEU, structural), so hypotheses that translate more entities correctly rise in the group ranking and propagate stronger updates, while those that omit or mistranslate entities receive negative advantages.

Overall, this localized supervision supplies crisp, high-signal feedback exactly where BLEU provides weak guidance: on entity fidelity. It reduces omissions, improves terminological consistency, and strengthens robustness on entity-dense biomedical or technical sentences.

## 5.4 Semantic Matching Reward

To encourage the model to attend to semantically relevant regions of the source—especially around named entities and their modifiers—we introduce an *Attention Similarity Reward*. For each hypothesis $\hat{y}$, we extract a target$\rightarrow$source attention distribution and compare it to a reference alignment $A^\star$ produced by an external aligner (e.g., `awesome-align`). In encoder–decoder models this is the cross-attention; in decoder-only models we form the concatenated sequence $[x \,\|\, \hat{y}]$ and take the self-attention *submatrix* from target positions to source positions.

**Attention extraction and aggregation.** Let $A^{(\ell,h)} \in \mathbb{R}^{T_{\text{tgt}} \times T_{\text{src}}}$ denote the head-$h$, layer-$\ell$ attention from target tokens to source tokens (or the corresponding submatrix for decoder-only). We aggregate heads/layers with nonnegative weights $w_{\ell,h}$ (either

uniform or tuned) and normalize row-wise to obtain a single stochastic matrix:

$$\bar{A} \;=\; \mathrm{Norm}_{\mathrm{row}}\Big(\sum_{\ell=1}^{L}\sum_{h=1}^{H} w_{\ell,h}\, A^{(\ell,h)}\Big) \in [0,1]^{T_{\mathrm{tgt}}\times T_{\mathrm{src}}}, \qquad \sum_{s=1}^{T_{\mathrm{src}}} \bar{A}_{t,s} = 1 \;\; \forall t. \tag{5.5}$$

Optionally, a temperature $\tau \in (0,\infty)$ can sharpen/soften $\bar{A}$ via $\mathrm{softmax}(\cdot/\tau)$, and a light Sinkhorn projection can make $\bar{A}$ approximately doubly stochastic for better alignment comparability.

**Similarity measure.** We score attention consistency with a length-invariant cosine similarity between the aggregated attention and the reference alignment:

$$R_{\mathrm{attn}} \;=\; \mathrm{sim}\big(\bar{A}, A^{\star}\big) \;=\; \frac{\langle \bar{A},\, A^{\star}\rangle_F}{\big\|\bar{A}\big\|_F \,\big\|A^{\star}\big\|_F + \varepsilon} \;\in\; [0,1], \tag{5.6}$$

where $\langle\cdot,\cdot\rangle_F$ is the Frobenius inner product and $\varepsilon > 0$ ensures numerical stability. This reward is high when the model places attention mass on the same source spans as the reference alignment.

**Integration with GRPO.** Within Group Relative Policy Optimization (GRPO), for a source $x$ we sample a group of $M$ hypotheses $\{\hat{y}_j\}_{j=1}^{M}$, compute per-sample attention rewards $r_j^{\mathrm{attn}} = \lambda_{\mathrm{attn}}\, R_{\mathrm{attn}}(x, \hat{y}_j)$, and center them with a groupwise baseline to obtain advantages

$$a_j^{\mathrm{attn}} \;=\; r_j^{\mathrm{attn}} \;-\; \frac{1}{M}\sum_{m=1}^{M} r_m^{\mathrm{attn}}. \tag{5.7}$$

These advantages feed the clipped GRPO/PPO objective to update $\pi_\theta$, naturally ranking hypotheses with better source-grounded attention higher and propagating stronger gradients.

Overall, this reward reduces semantic drift, encourages source-anchored decisions, and improves robustness in entity-dense biomedical and technical sentences by aligning the model's attention with linguistically plausible token correspondences.

## 5.5 Syntactic Matching Reward

To preserve relational meaning beyond surface $n$-gram overlap, we introduce a *Global Syntactic Graph Similarity (GSGS) reward* that evaluates the hypothesis $\hat{y}$ against the *reference* sentence $y^{\star}$ at the level of sentence structure. The reward admits two interchangeable implementations that compute global syntactic similarity using either (i) *dependency* trees or (ii) *constituency (CFG)* trees for both $y^{\star}$ and $\hat{y}$. In both cases we compare labeled graphs that encode who-governs-whom and how modifiers, arguments, and clausal links are arranged—precisely the scaffolding that carries roles (agent/patient), scope (e.g., negation), and clinically crucial qualifiers (dose, units, temporality).

**Dependency-graph variant.** For each sentence $y$, we build a labeled, directed dependency graph $G_{\mathrm{dep}}(y)$ whose nodes carry composite labels (e.g., lemma/POS/deprel, optionally augmented with entity tags), and edges encode head→dependent relations.

After $k$ rounds of Weisfeiler–Lehman (WL) refinement, we compute a length- and scale-invariant similarity:

$$R_{\mathrm{dep}}(y^\star, \hat{y}) = \widetilde{K}_{\mathrm{WL}}\big(G^{(k)}_{\mathrm{dep}}(y^\star),\, G^{(k)}_{\mathrm{dep}}(\hat{y})\big) = \frac{K_{\mathrm{WL}}\big(G^{(k)}_{\mathrm{dep}}(y^\star),\, G^{(k)}_{\mathrm{dep}}(\hat{y})\big)}{\sqrt{K_{\mathrm{WL}}\big(G^{(k)}_{\mathrm{dep}}(y^\star),\, G^{(k)}_{\mathrm{dep}}(y^\star)\big)\, K_{\mathrm{WL}}\big(G^{(k)}_{\mathrm{dep}}(\hat{y}),\, G^{(k)}_{\mathrm{dep}}(\hat{y})\big)}} \;\in\; [0,1]. \tag{5.8}$$

Small $k$ (e.g., $k \in \{2,3\}$) captures most linguistically relevant neighborhoods while remaining robust to parser noise.

**CFG-graph variant.** Alternatively, we parse both sentences into constituency trees $T_{\mathrm{cfg}}(y)$ (nonterminal labels, head-word or POS annotations as desired) and view each tree as a labeled, directed graph $G_{\mathrm{cfg}}(y)$ via parent↔child edges. We then compute an analogous normalized similarity—either via a tree kernel $K_{\mathrm{CFG}}$ or by applying WL refinement to $G_{\mathrm{cfg}}$:

$$R_{\mathrm{cfg}}(y^\star, \hat{y}) \;=\; \widetilde{K}_{\mathrm{CFG}}\Big(G_{\mathrm{cfg}}(y^\star),\, G_{\mathrm{cfg}}(\hat{y})\Big) \;\in\; [0,1]. \tag{5.9}$$

This variant emphasizes phrasal configuration and long-distance constituent structure, complementing the head–dependent view of dependencies.

**Integration with GRPO.** Within Group Relative Policy Optimization, for each source $x$ we sample a group of $M$ hypotheses $\{\hat{y}_j\}_{j=1}^{M}$ and compute per-sample structural rewards

$$r_j^{\mathrm{syn}} \;=\; \lambda_{\mathrm{syn}}\, R_{\mathrm{syn}}(y^\star, \hat{y}_j), \qquad a_j \;=\; r_j^{\mathrm{syn}} - \frac{1}{M}\sum_{m=1}^{M} r_m^{\mathrm{syn}}, \tag{5.10}$$

then use $\{a_j\}$ in the clipped GRPO objective (PPO-style). Hypotheses with more faithful global syntax receive higher advantages, providing a stabilizing structural pressure that counters meaning drift, preserves entity roles and scopes, and improves clinical fidelity. On parsing failures or low-confidence parses, we assign a neutral default (or smoothly down-weight via $\lambda_{\mathrm{syn}}$) to maintain stable training.

## 5.6 Experimental Setup and Results

**Dataset.** We conduct fine-tuning on the MedEV corpus: the *training* split is used to update model weights; the *development* split is used for hyperparameter selection and early stopping; and the span-annotated *test* split introduced in Chapter 3 serves as the held-out benchmark for final evaluation. All experiments are run in the translation from English to Vietnamese, matching the reporting in Table 5.1.

**Model.** Our experiments are conducted using *Qwen2.5-7B-Instruct*, which was selected for its favorable balance between computational efficiency and model capability. With approximately 7 billion parameters, the model enables efficient training and inference on commodity GPUs while maintaining strong instruction-following behavior and high-quality multilingual translation performance. Reinforcement learning fine-tuning is implemented within the Hugging Face Transformers ecosystem [53]. To improve parameter efficiency, we employ Low-Rank Adaptation (LoRA) [54] and apply it to the

query ($Q$), key ($K$), and value ($V$) projection matrices of the self-attention layers. All reinforcement learning variants are trained for three epochs using a learning rate of $1 \times 10^{-5}$ and the `bfloat16` precision format.

To incorporate syntactic information into the reward design, dependency and constituency (CFG) parse trees are generated using the Stanza toolkit [55]. The resulting dependency graphs and CFG trees are subsequently used to compute the proposed graph-based syntactic similarity rewards. We compare the original pretrained model against multiple reinforcement learning variants optimized with different combinations of lexical, entity, semantic, and syntactic reward functions, corresponding to the configurations marked with "*" in Table 5.1.

For baseline comparisons, we consider both the original *Qwen2.5-7B-Instruct* model and a supervised fine-tuning (SFT) variant. The SFT model is trained using LoRA with a rank of 64, a learning rate of $1 \times 10^{-4}$, and a training duration of five epochs. These baselines enable us to assess the contribution of reinforcement learning beyond standard parameter-efficient fine-tuning and to quantify the impact of the proposed reward components on translation quality, entity preservation, semantic fidelity, and syntactic consistency.

**Evaluation Metrics.** We report corpus-level BLEU [25] and the METEOR score[26] as the measures of overall translation quality and M-ETA [40] to quantify entity-level translation fidelity. Besides, we also report BERTScore[27] for the semantic preservation in the translation result. For all metrics, higher scores indicate better performance, as reflected by the ↑ markers in Table 5.1 and in Table 5.2. During the fine-tuning process, we also observe the change in loss value and the mean value of every reward function in that combination.

| **Model / Setting** | **BLEU ↑** | | **M-ETA ↑** | | **METEOR ↑** | |
|---|---|---|---|---|---|---|
| | **Eval** | **Test** | **Eval** | **Test** | **Eval** | **Test** |
| Qwen2.5 7B Instruct (vanilla) | 23.44 | 22.05 | 33.72 | 32.63 | 55.28 | 53.78 |
| Qwen2.5 7B Instruct (LorA SFT fine-tuning) | 27.78 | 27.50 | 33.83 | 32.92 | 58.48 | 58.38 |
| Qwen2.5 7B Instruct (RL fine-tuning Lex+Ent) * | 36.22 | 36.08 | 36.94 | 37.94 | 61.25 | 60.38 |

Table 5.1: Comparison of best performance of our framework (Lexical Matching + Entity Matching RL) with baselines on BLEU score, M-ETA score, and METEOR score.

## 5.7 Dicussion and Conclusion

RL fine-tuning yields large gains over the SFT fine-tuned baseline: BLEU rises from 27.50 to 36.08 on the test set (+8.58), and M-ETA improves from 32.92 to 37.94 when the reward combines Lexical Matching and Entity Matching, indicating that entity-aware feedback enhances both global fluency and entity fidelity. Using Lexical Matching alone already improves BLEU substantially (34.90) but raises M-ETA only marginally (34.96), suggesting Entity Matching is the key driver of entity correctness. In contrast, attention-based rewards help improve the performance of the model

| Model / Setting | P ↑ | | R ↑ | | F1 ↑ | |
|---|---|---|---|---|---|---|
| | **Eval** | **Test** | **Eval** | **Test** | **Eval** | **Test** |
| Qwen2.5 7B Instruct (vanilla) | 76.36 | 75.86 | 83.16 | 82.69 | 79.15 | 78.59 |
| Qwen2.5 7B Instruct (LorA SFT fine-tuning) | 81.57 | 81.36 | 85.46 | 85.31 | 83.21 | 83.03 |
| Qwen2.5 7B Instruct (RL fine-tuning Lex+Ent) * | 86.09 | 85.76 | 86.48 | 86.06 | 86.25 | 85.86 |

Table 5.2: Comparison of best performance of our framework (Lexical Matching + Entity Matching RL) with baselines on BERTScore.

| RL Reward Setting | BLEU ↑ | | M-ETA ↑ | | METEOR ↑ | |
|---|---|---|---|---|---|---|
| | **Eval** | **Test** | **Eval** | **Test** | **Eval** | **Test** |
| Lex | 35.11 | 34.90 | 35.51 | 34.96 | 59.75 | 58.97 |
| Lex+Ent | 36.22 | 36.08 | 36.94 | 37.94 | 61.25 | 60.38 |
| Lex+Ent+Sem | 37.23 | 35.90 | 38.38 | 35.24 | 62.54 | 61.05 |
| Lex+Ent+DepSyn | 35.85 | 34.13 | 37.61 | 37.37 | 61.13 | 60.51 |
| Lex+Ent+CFGSyn | 35.41 | 33.73 | 36.11 | 34.77 | 60.23 | 58.75 |
| Lex+Ent+Sem+CFGSyn | 36.27 | 34.98 | 36.94 | 35.99 | 60.02 | 58.83 |
| Lex+Ent+Sem+DepSyn | 35.99 | 34.51 | 37.94 | 37.16 | 61.60 | 60.11 |
| Lex+DepSyn | 35.83 | 34.81 | 36.48 | 35.28 | 61.32 | 60.45 |

Table 5.3: Ablation study: BLEU and M-ETA scores for different RL reward configurations.

| RL Reward Setting | BLEU ↑ | | M-ETA ↑ | | METEOR ↑ | |
|---|---|---|---|---|---|---|
| | **Eval** | **Test** | **Eval** | **Test** | **Eval** | **Test** |
| Lex | 85.19 | 84.80 | 85.87 | 85.51 | 85.47 | 85.09 |
| Lex+Ent | 86.09 | 85.76 | 86.48 | 86.06 | 86.25 | 85.86 |
| Lex+Ent+Sem | 86.35 | 85.90 | 86.89 | 86.27 | 86.58 | 86.05 |
| Lex+Ent+DepSyn | 85.21 | 85.20 | 86.24 | 86.12 | 85.67 | 85.61 |
| Lex+Ent+CFGSyn | 85.15 | 84.70 | 86.37 | 85.88 | 85.71 | 85.24 |
| Lex+Ent+Sem+CFGSyn | 84.71 | 84.67 | 85.59 | 85.29 | 85.09 | 84.93 |
| Lex+Ent+Sem+DepSyn | 86.08 | 85.71 | 86.83 | 86.33 | 86.41 | 85.98 |
| Lex+DepSyn | 86.10 | 85.87 | 86.73 | 86.40 | 86.37 | 86.09 |

Table 5.4: Ablation study: BERTScore for different RL reward configurations.

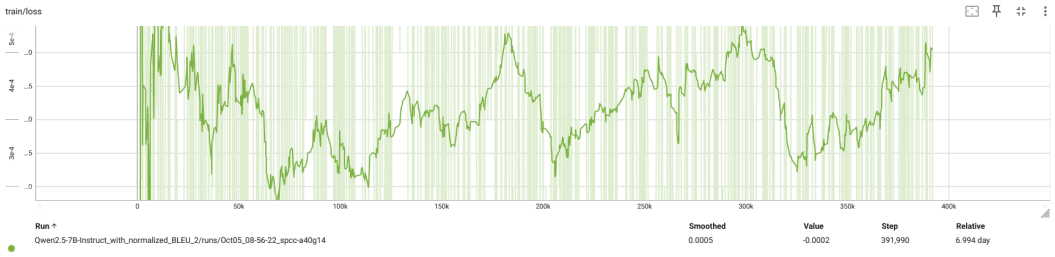


Figure 5.2: The loss of the training process while applying only the lexical matching reward function in the GRPO algorithm.

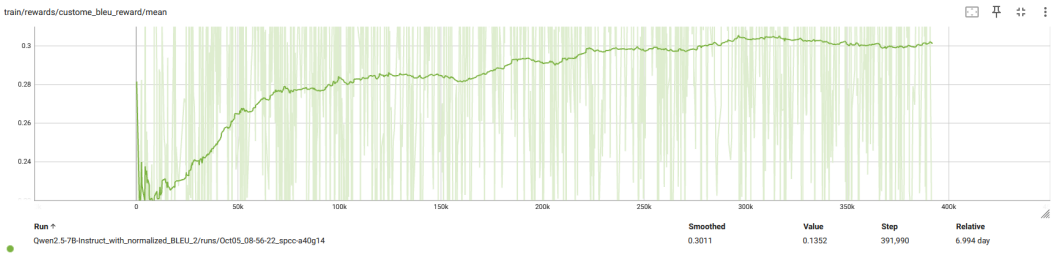


Figure 5.3: The mean value of BLUE score of the model on the evaluation dataset during the training process while applying only the lexical matching reward function in the GRPO algorithm.

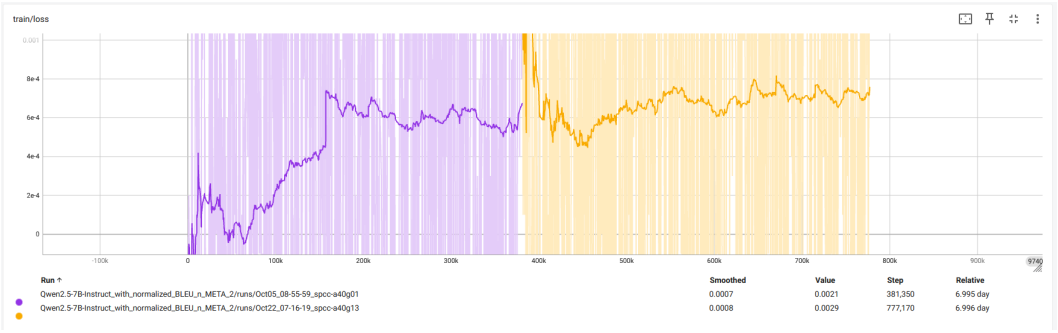


Figure 5.4: The loss of the training process while applying the combination between the lexical matching reward function and entity matching reward function in the GRPO algorithm.

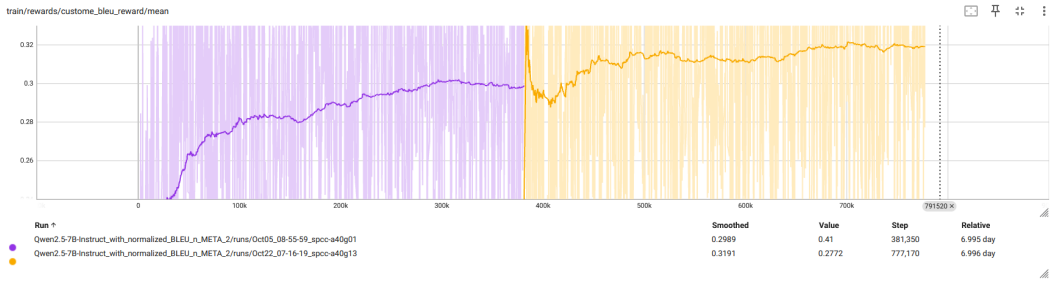


Figure 5.5: The mean value of BLUE score of the model on the evaluation dataset during the training process while applying the combination between the lexical matching reward function and entity matching reward function in the GRPO algorithm.

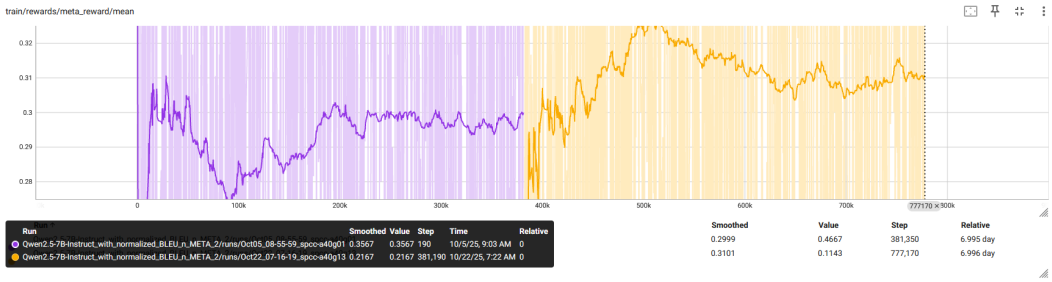


Figure 5.6: The mean value of the M-ETA score of the model on the evaluation dataset during the training process while applying the combination between the lexical matching reward function and entity matching reward function in the GRPO algorithm.

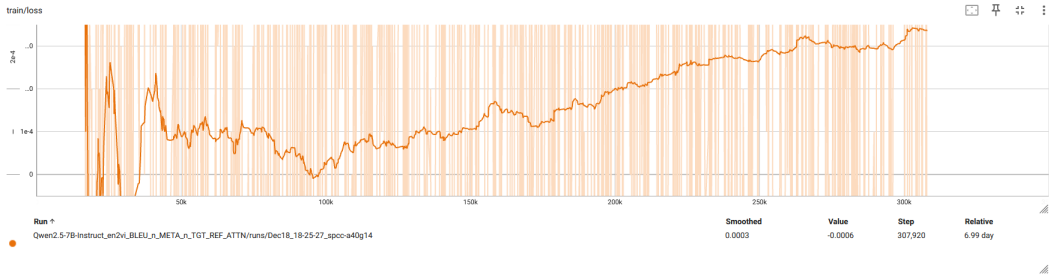


Figure 5.7: The loss of the training process while applying three reward functions of lexical matching, entity matching, and semantic matching.

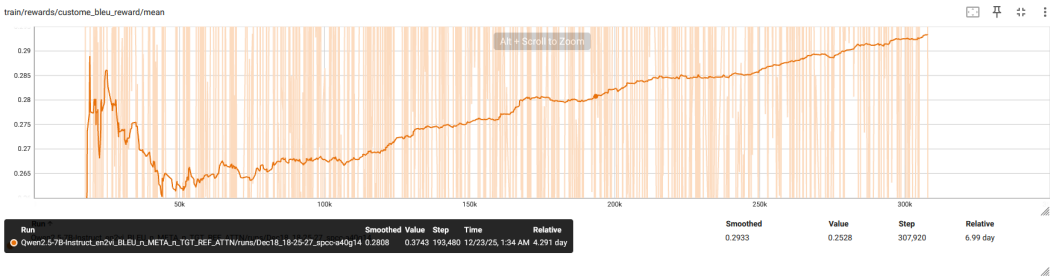


Figure 5.8: The mean of BLEU score during the training process while applying three reward functions of lexical matching, entity matching, and semantic matching.

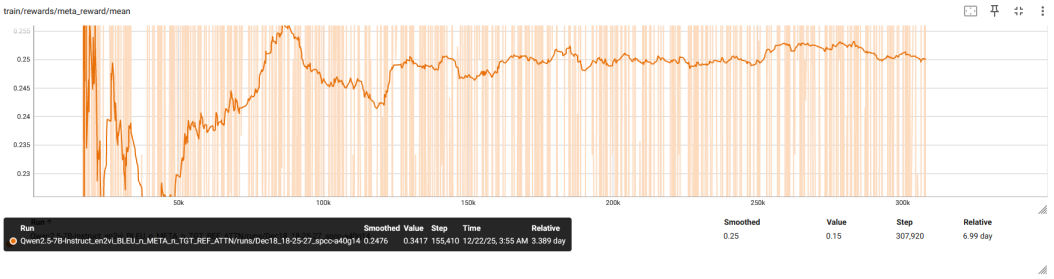


Figure 5.9: The mean of M-ETA score during the training process while applying three reward functions of lexical matching, entity matching, and semantic matching.

strongly in the validation dataset. However, in the experiment with the test dataset, the performance is lower compared with the setting when the reward combines Lexical Matching and Entity Matching. This shows that an attention-matching reward has

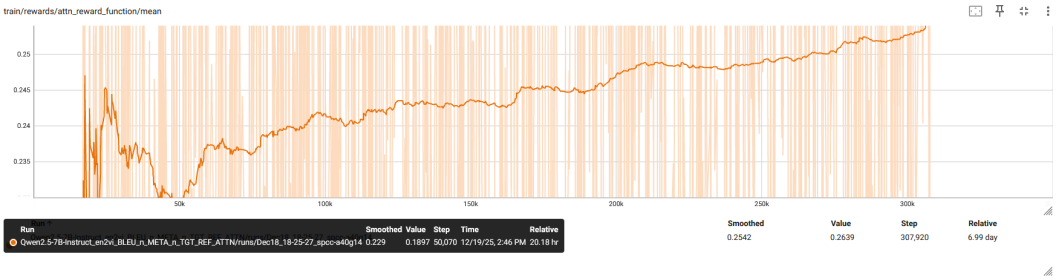


Figure 5.10: The mean of attention reward score during the training process while applying three reward functions of lexical matching, entity matching, and semantic matching.

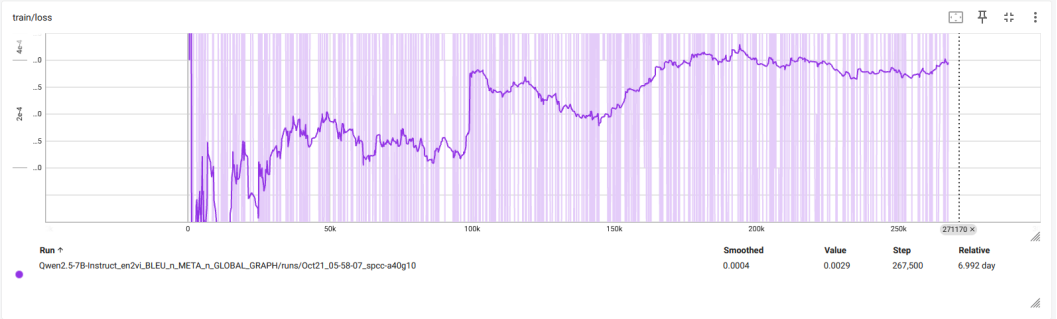


Figure 5.11: The loss of the training process when combining the lexical matching reward, entity matching reward, and syntactic matching reward. In this case, the syntactical matching reward function is implemented using the similarity between the dependency graphs of the reference sentences and the translated sentences.

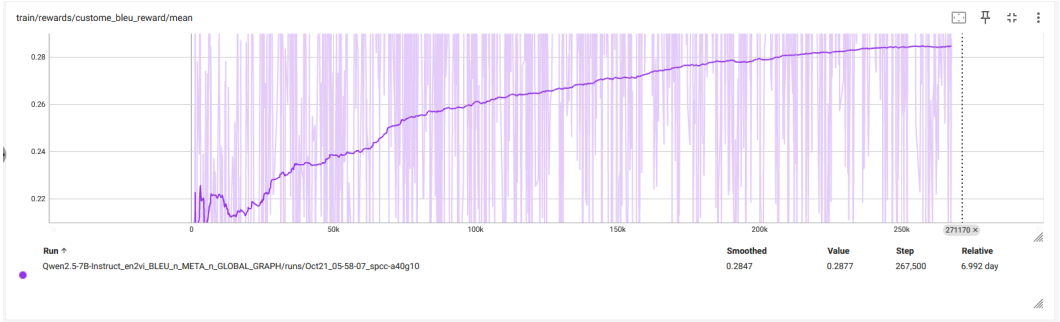


Figure 5.12: The mean value of the BLEU score during the fine-tuning process when combining the lexical matching reward, entity matching reward, and syntactic matching reward. In this case, the syntactical matching reward function is implemented using the similarity between the dependency graphs of the reference sentences and the translated sentences.

high potential in boosting the performance of the framework, but can be harmed when meeting new data. With the full stack of *Lex+Ent+Sem+DepSyn*, the results show an imbalance trade-off to boost the performance in entity translation and a slight improvement in the BLEU score. The combination of syntactic similarity reward seems to make a strong contribution in guiding the model to translate entities correctly, but greatly reduces the BLEU score compared to other combinations.

When evaluated using the METEOR metric, the reward combination consisting of

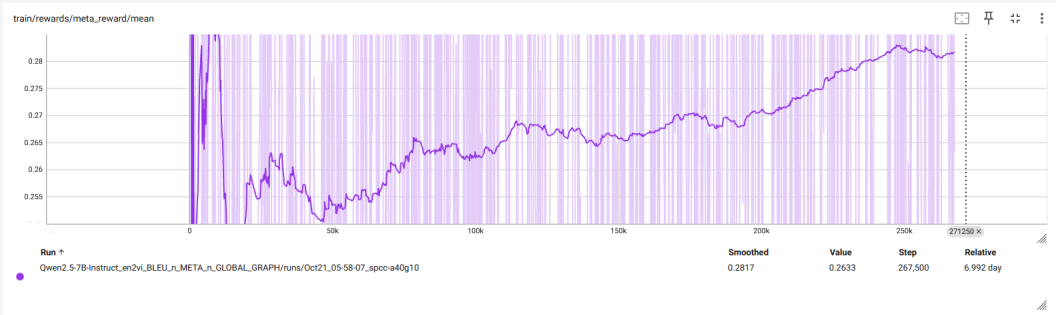


Figure 5.13: The mean value of the M-ETA score during the fine-tuning process when combining the lexical matching reward, entity matching reward, and syntactic matching reward. In this case, the syntactical matching reward function is implemented using the similarity between the dependency graphs of the reference sentences and the translated sentences.

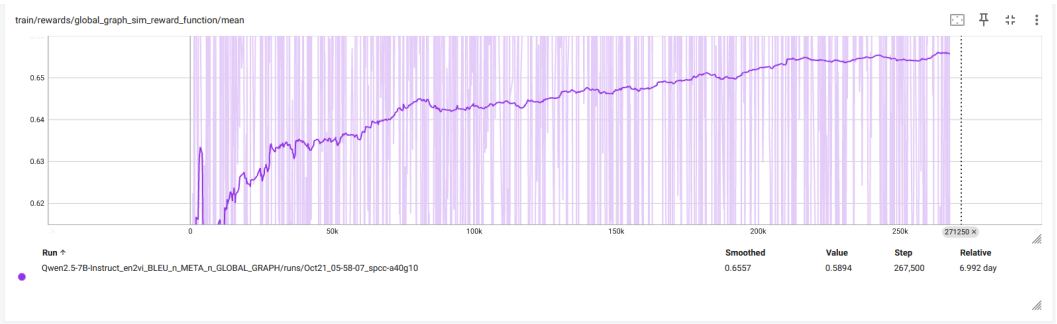


Figure 5.14: The mean value of the attention similarity score during the fine-tuning process when combining the lexical matching reward, entity matching reward, and syntactic matching reward. In this case, the syntactical matching reward function is implemented using the similarity between the dependency graphs of the reference sentences and the translated sentences.

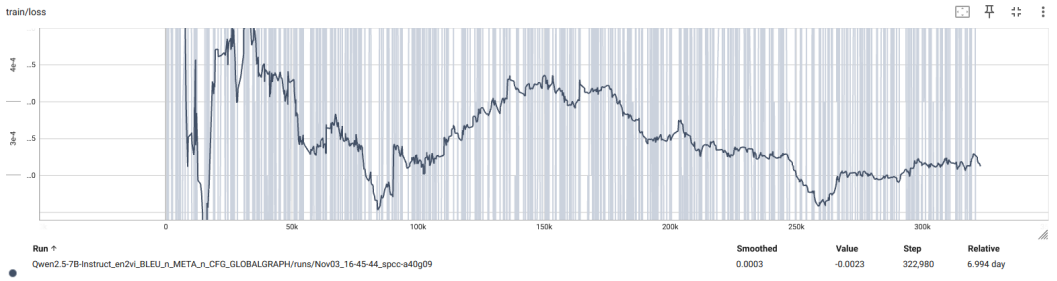


Figure 5.15: The loss of the training process when combining the lexical matching reward, entity matching reward, and syntactic matching reward. In this case, the syntactical matching reward function is implemented using the similarity between the CFG trees of the reference sentences and the translated sentences.

*Lexical Matching*, *Entity Matching*, and *Semantic Matching* achieves the strongest performance among all investigated configurations. Specifically, it not only significantly outperforms the supervised fine-tuning (SFT) LoRA baseline but also consistently surpasses alternative reinforcement learning reward combinations. This result highlights the effectiveness of incorporating semantic-level feedback into the optimization pro-

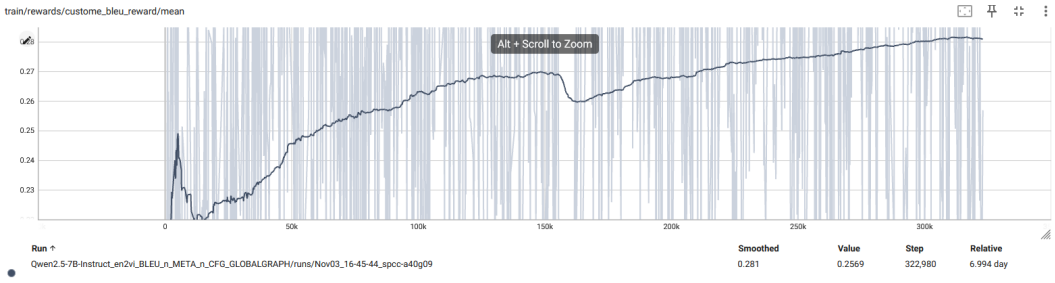


Figure 5.16: The mean value of the BLEU score during the fine-tuning process when combining the lexical matching reward, entity matching reward, and syntactic matching reward. In this case, the syntactical matching reward function is implemented using the similarity between the CFG trees of the reference sentences and the translated sentences.

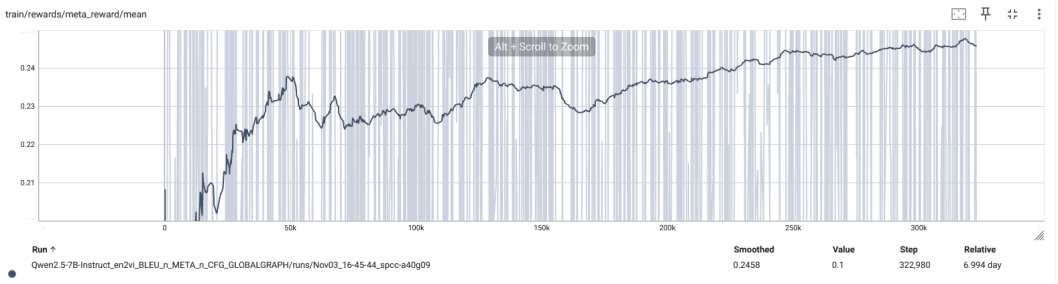


Figure 5.17: The mean value of the M-ETA score during the fine-tuning process when combining the lexical matching reward, entity matching reward, and syntactic matching reward. In this case, the syntactical matching reward function is implemented using the similarity between the CFG trees of the reference sentences and the translated sentences.

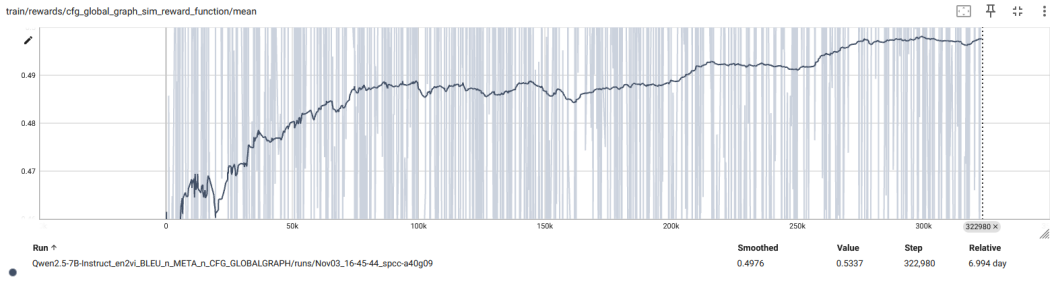


Figure 5.18: The mean value of the attention similarity score during the fine-tuning process when combining the lexical matching reward, entity matching reward, and syntactic matching reward. In this case, the syntactical matching reward function is implemented using the similarity between the CFG trees of the reference sentences and the translated sentences.

cess. While lexical and entity-based rewards encourage accurate word selection and preservation of important named entities, the Semantic Matching reward provides a complementary signal that guides the model toward generating translations that better preserve the underlying meaning of the source sentence. As a result, the model is able to produce translations that are not only lexically correct but also semantically faithful to the original content.

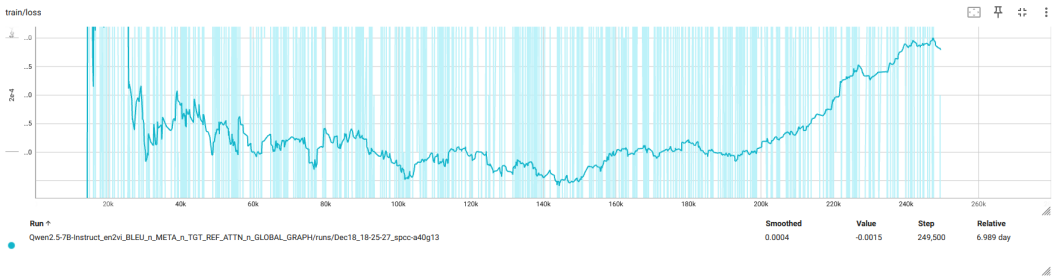


Figure 5.19: The loss of the training process when combining all 4 of the reward functions. In this case, the syntactical matching reward function is implemented using the similarity between the dependency graphs of the reference sentences and the translated sentences.

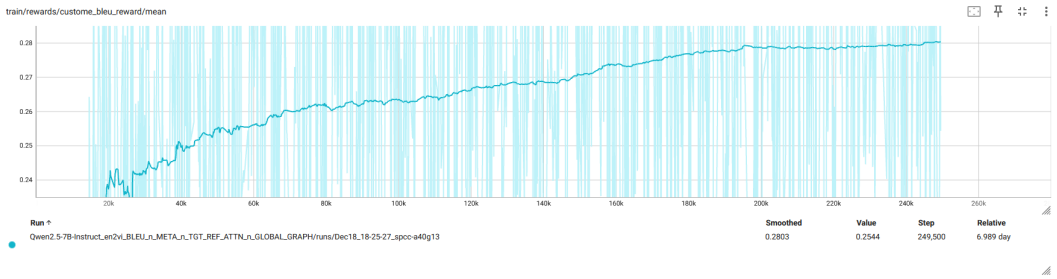


Figure 5.20: The mean value of the BLEU score during the fine-tuning process when combining all 4 reward functions. In this case, the syntactical matching reward function is implemented using the similarity between the dependency graphs of the reference sentences and the translated sentences.

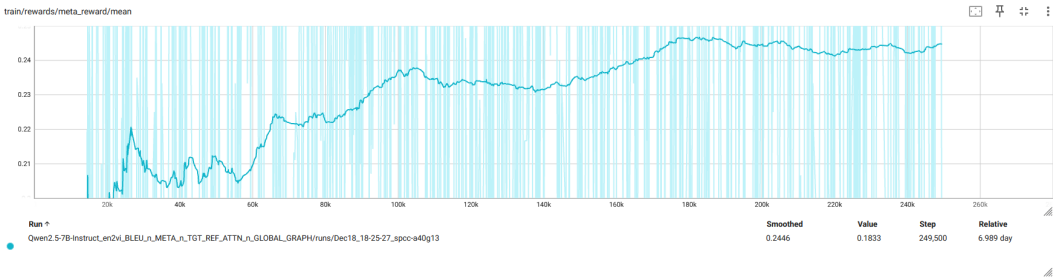


Figure 5.21: The mean value of the M-ETA score during the fine-tuning process when combining all 4 reward functions. In this case, the syntactical matching reward function is implemented using the similarity between the dependency graphs of the reference sentences and the translated sentences.

A similar trend can be observed when evaluating the models using BERTScore. The configuration that includes the Semantic Matching reward again achieves the highest performance, demonstrating its substantial contribution to semantic preservation during translation. Since BERTScore measures similarity in the contextual embedding space rather than relying solely on surface-level token overlap, its improvement indicates that the generated translations capture the intended meaning more effectively. The consistent gains observed across both METEOR and BERTScore suggest that the Semantic Matching reward plays a crucial role in enhancing translation quality by encouraging the model to maintain semantic equivalence between the source and target texts. These findings further indicate that semantic-aware reward signals

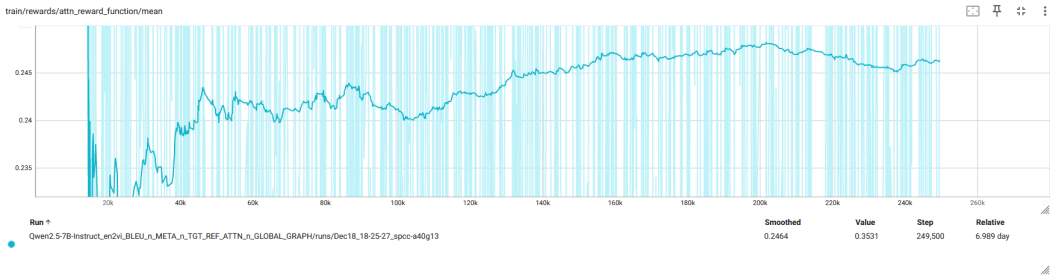


Figure 5.22: The mean value of the attention similarity score during the fine-tuning process when combining all 4 reward functions. In this case, the syntactical matching reward function is implemented using the similarity between the dependency graphs of the reference sentences and the translated sentences.

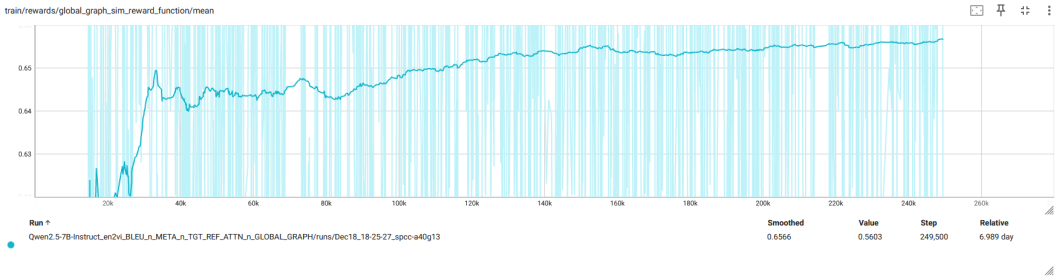


Figure 5.23: The mean value of the dependency graph similarity score during the fine-tuning process when combining all 4 reward functions. In this case, the syntactical matching reward function is implemented using the similarity between the dependency graphs of the reference sentences and the translated sentences.

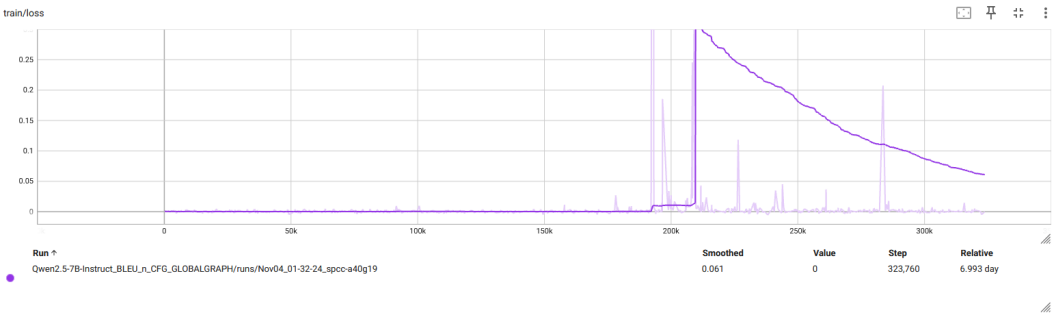


Figure 5.24: The loss during the fine-tuning process when combining lexical matching reward and syntactic matching reward.

are particularly valuable for reinforcement learning-based machine translation, as they help address limitations of purely lexical objectives and promote translations that align more closely with human judgments of adequacy and meaning preservation.

**Loss and mean-reward trajectories.** Under GRPO, training *loss* generally declines while the *mean value* of the optimized reward(s) rises; the precise dynamics depend on the reward mix:

- **Lexical Matching only.** Loss decreases smoothly and mean BLEU grows monotonically. This is the most stable setting, reflecting a strong lexical signal; entity-focused metrics (e.g., M-ETA) remain largely flat because they are not directly optimized.

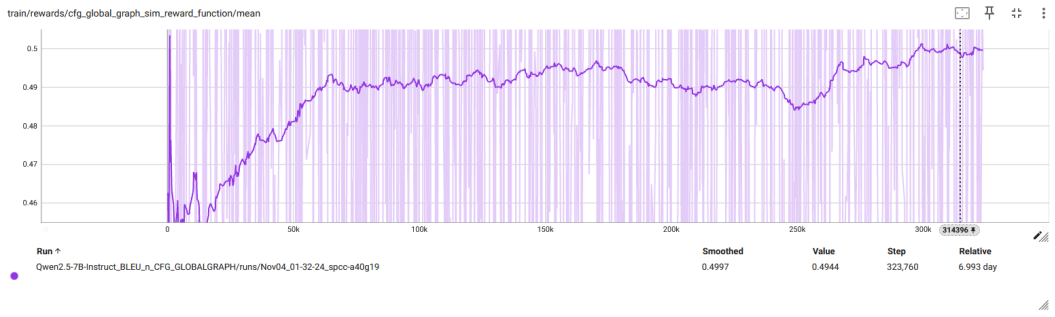


Figure 5.25: The mean value of the BLEU score during the fine-tuning process when combining lexical matching reward and syntactic matching reward.

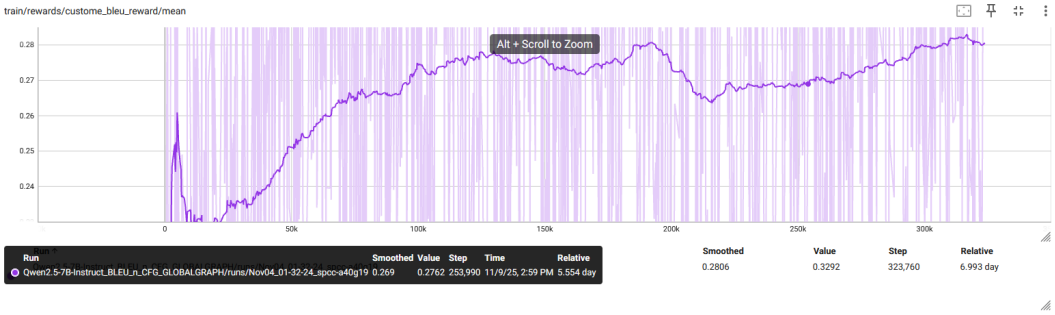


Figure 5.26: The mean value of the dependency graph similarity score during the fine-tuning process when combining lexical matching reward and syntactic matching reward.

- **Lexical Matching + Entity Matching.** Loss drops steadily while both mean BLEU and mean M-ETA increase throughout training. This pairing offers the best overall balance: fluency improves without sacrificing entity fidelity, and the two curves co-improve with no evident reward conflict.
- **Lexical Matching + Entity Matching + Semantic Matching.** Optimization becomes more sensitive: some runs exhibit noisier or partially non-convergent loss. The mean semantic score climbs, but gains in BLEU and M-ETA are shallower, indicating a credit-assignment tension in which the semantic signal competes with lexical/entity objectives.
- **Lexical Matching + Entity Matching + Syntactic Matching (Dependency graph).** Loss declines reliably; mean BLEU and mean M-ETA rise, and the mean dependency-graph similarity increases throughout training. The structural signal stabilizes word order and relations with only a modest BLEU trade-off relative to the Lexical+Entity best point.
- **Lexical Matching + Entity Matching + Syntactic Matching (CFG graph).** Dynamics mirror the dependency variant: decreasing loss accompanied by concurrent increases in mean BLEU, mean M-ETA, and CFG-based graph similarity. As above, the structure-aware reward slightly tempers BLEU while preserving (or modestly improving) entity fidelity.
- **Lexical Matching + Entity Matching + Semantic Matching + Syntactic Matching.** With all four signals, loss reduction is weaker and can fluctuate. Mean

semantic and graph similarities rise, but mean BLEU and mean M-ETA lag behind simpler setups, consistent with reward competition and less stable gradients when optimizing many objectives simultaneously.

**Takeaway.** *Lexical-only* training is the most stable; *Lexical + Entity* delivers the best quality balance (fluency & entity correctness); *syntactic* rewards (dependency/CFG) are supportive but introduce a small BLEU trade-off; and adding *semantic* on top of lexical+entity requires careful weighting/curriculum to avoid dampening BLEU/M-ETA and complicating loss convergence.

## 5.8 Summary of the Chapter

This chapter introduced a *Multi-Aspect Reinforcement Learning* framework for entity-aware machine translation trained with **Group Relative Policy Optimization (GRPO)**. We framed the translator as a policy that samples groups of hypotheses and learns by ranking candidates with respect to a composite reward. Four complementary rewards were defined: (i) a *Lexical Matching* signal based on sentence-level BLEU to anchor global fluency, (ii) an *Entity Matching* signal (M-ETA) to enforce correct rendering of clinically salient terms, (iii) a *Semantic Matching* signal to encourage semantic consistency between source and target sentences, and (iv) a *Syntactic Matching* signal (*GSGS*) that measures global structural similarity via either dependency or CFG graphs. The GRPO formulation reduced sensitivity to absolute reward scales and eliminated the need for a KL-anchored reference policy by computing within-group advantages. By jointly optimizing lexical, entity-level, semantic, and structural objectives, the proposed framework aims to improve not only translation fluency and terminology accuracy but also meaning preservation and syntactic faithfulness.

Empirically, we fine-tuned *Qwen2.5-7B-Instruct* on MedEV and evaluated the resulting models using both traditional and semantic-oriented translation metrics, including BLEU, M-ETA, METEOR, and BERTScore. When considering BLEU and M-ETA, the combination of **Lexical Matching** and **Entity Matching** delivered the best balance between translation fluency and entity fidelity (e.g., Test BLEU $\approx 36.08$, M-ETA $\approx 37.94$), substantially outperforming both the vanilla baseline (BLEU 22.05, M-ETA 32.63) and the SFT fine-tuned baseline (BLEU 27.50, M-ETA 32.92). Lexical Matching-only training exhibited the most stable optimization behavior but yielded only limited improvements in entity accuracy. Furthermore, incorporating attention-based rewards often destabilized training and resulted in lower BLEU and M-ETA scores. Structure-aware rewards based on dependency or CFG graphs generally improved structural consistency and helped preserve entity correctness, although at the expense of a slight reduction in BLEU performance.

A different pattern emerged when evaluating semantic quality using METEOR and BERTScore. The reward combination consisting of **Lexical Matching**, **Entity Matching**, and **Semantic Matching** achieved the strongest results across both metrics, outperforming not only the SFT baseline but also all other reinforcement learning reward configurations. The consistent gains observed in METEOR indicate that semantic-aware optimization improves the model's ability to preserve lexical meaning while maintaining accurate terminology translation. More importantly, the superior BERTScore results demonstrate that the Semantic Matching reward effectively encourages translations that remain semantically faithful to the source text, even when alternative lexical realizations are used. Since BERTScore evaluates contextual semantic similarity rather

than surface-level token overlap, these improvements suggest that the generated translations better capture the intended meaning of the original sentences. Together, the METEOR and BERTScore results provide strong evidence that semantic-level reward signals contribute substantially to meaning preservation and complement lexical and entity-oriented objectives.

Overall, the experimental findings validate the central claim of this chapter: *multi-aspect* reinforcement learning can effectively improve medical machine translation by targeting multiple dimensions of translation quality simultaneously. While the combination of lexical and entity rewards provides the strongest improvements in fluency and terminology accuracy, the addition of semantic rewards yields further gains in semantic preservation as measured by METEOR and BERTScore. Moreover, syntactic rewards offer a useful regularization mechanism for maintaining structural consistency without significantly compromising entity fidelity. These results highlight the importance of balancing lexical, entity-level, semantic, and structural objectives within a unified reinforcement learning framework. The chapter concludes by providing practical insights into reward design and optimization strategies, and motivates subsequent analyses on broader evaluation settings and ablation studies.

# Chapter 6

# Conclusion

## 6.1 Summary of Contributions

This thesis addressed two intertwined challenges in entity-aware machine translation for terminology-sensitive domains: the scarcity of high-quality, entity-annotated training and evaluation data, and the problem of catastrophic forgetting that arises when adapting a general-domain translation model to a specialized domain. Rather than treating entity fidelity and general-domain fluency as competing objectives, we proposed a unified framework that integrates them through multi-aspect reinforcement learning with Group Relative Policy Optimization (GRPO). The key contributions are summarized as follows.

**Multi-task entity-aware translation model (Chapter 3).** We developed a multi-task learning framework that integrates named entity recognition signals directly into an mT5-based translation pipeline. By explicitly marking entity spans with lightweight boundary tags and training the model to first recognize and translate entities before generating the full sentence, the approach foregrounds entity-related information during decoding. Experiments on the SemEval 2025 Task 2 benchmark demonstrated consistent improvements over both mBART and a standard mT5 baseline across four language pairs (English-German, English-French, English-Italian, English-Spanish), confirming that jointly optimizing NER and translation within a unified architecture strengthens entity-level accuracy.

**Entity-annotated dataset construction (Chapter 4).** To address the data bottleneck, we constructed an enriched version of the MedEV English-Vietnamese test split with explicit, bilingual span-level annotations for clinically salient entities (diseases, anatomical structures, procedures, drugs). The annotation framework is grounded in authoritative sources,MedlinePlus (215 terms), the Multilingual Cancer Glossary (612 entries), and Vietnamese Ministry of Health publications (44000 standardized terms), and governed by a set of consistency policies covering compound-mention decomposition, abbreviation handling, brand-versus-generic drug normalization, and synonym canonicalization. Across the MedEV test set of 8,960 sentence pairs, we annotated 3,675 pairs (41.0%), producing 25,308 source-target entity annotations. To scale annotation to the training corpus, we designed a four-stage automatic pipeline,NER extraction, LLM-based alignment, sentence-pair mining, and merge-and-filter, that produces a silver-standard dataset mirroring the structure and constraints of the manually cu-

rated test set. A manual evaluation of 200 sampled sentence pairs yielded a precision of 64.86%, with an accompanying error analysis that categorized the primary failure modes (boundary errors, generic terms, abbreviation mismatches, procedural nominalizations, and prognostic statements).

**Multi-aspect reinforcement learning framework (Chapter 5).** We proposed a novel multi-aspect RL framework that formulates entity-aware translation as a GRPO problem, where the translation model acts as a policy that samples groups of candidate hypotheses and learns by ranking them with respect to a composite reward. Four complementary reward functions were defined and formalized: (i) a *Lexical Matching* reward based on sentence-level BLEU to anchor global fluency, (ii) an *Entity Matching* reward (M-ETA) to enforce correct rendering of domain-specific terms, (iii) a *Semantic Matching* reward derived from attention alignment to encourage source-grounded generation, and (iv) a *Syntactic Matching* reward based on Weisfeiler–Lehman graph kernel similarity over dependency or constituency parse trees to preserve relational structure. The GRPO formulation eliminates the need for a KL-anchored reference policy and reduces sensitivity to absolute reward scales by computing within-group, rank-based advantages.

**Systematic ablation and empirical validation (Chapter 5).** Fine-tuning Qwen2.5-7B-Instruct on the MedEV corpus with different reward combinations revealed clear and actionable patterns. The combination of Lexical Matching and Entity Matching delivered the best balance between fluency and entity fidelity, yielding test-set BLEU of approximately 36.08 and M-ETA of approximately 37.94, substantially outperforming both the vanilla baseline (BLEU 22.05, M-ETA 32.63) and the SFT fine-tuned baseline (BLEU 27.50, M-ETA 32.92). When evaluated on semantic preservation via METEOR and BERTScore, the addition of Semantic Matching further improved performance, demonstrating that semantic-level feedback complements lexical and entity-oriented objectives. Syntactic rewards based on dependency or CFG graphs improved structural consistency and entity fidelity at the cost of a modest BLEU reduction, offering a useful regularization mechanism. Attention-based rewards, while improving validation-set scores, degraded generalization to the test set, indicating the need for more robust semantic reward designs.

## 6.2 Limitations

Despite the contributions outlined above, several limitations should be acknowledged.

- **Annotation coverage and alignment.** The reference lexicon provides strong supervision but remains incomplete, rare diseases, pediatric terms, device and brand names, legacy Vietnamese spellings, and evolving oncology nomenclature are underrepresented. Span boundaries can drift in edge cases involving dose attachments, stage and grade tokens, hyphenated forms, and parenthetical aliases. English-Vietnamese reordering patterns and generic-to-specific mappings may yield correct translations that the strict lexicon does not list verbatim, creating false negatives in evaluation.
- **Silver-set construction quality.** Automatic NER still produces false positives and false negatives, particularly in Vietnamese, and imperfect category alignment

with the target taxonomy. LLM-based alignment and mining can hallucinate canonical forms, over-normalize brand names to generics, or degrade on long or heavily reordered sentences. The merge-and-filter thresholds trade recall against precision and may either discard long-tail entities or admit subtle mismatches.

- **Learning dynamics and reward design.** Multi-reward training can be unstable; certain runs exhibited non-convergent loss with degraded outcomes. Attention-based (semantic) rewards improved validation-set performance but consistently reduced test-set quality, indicating overfitting or reward misalignment. Global graph-based syntactic rewards preserved entity scores but tended to reduce fluency, suggesting brittle credit assignment when structural constraints are enforced too broadly.

- **Scope of evaluation.** All RL experiments were conducted on a single language pair (English-Vietnamese) and a single domain (medical). While the multi-task NMT model in Chapter 3 was evaluated on four European language pairs, the generalizability of the RL framework to other languages, domains, and model scales remains to be established.

## 6.3 Future Work

Several promising directions emerge from the findings and limitations of this thesis.

- **Data and evaluation refinement.** Expand and periodically refresh the bilingual medical lexicon to improve coverage of under-represented entity categories. Refine boundary policies for doses, stages, hyphenations, and acronyms. Adopt alignment-tolerant adjudication strategies, such as one-to-many mappings and generic-to-specific normalization, to reduce false negatives on legitimate English-Vietnamese translation variants.

- **Silver-set robustness.** Improve in-domain Vietnamese NER through targeted fine-tuning on medical corpora. Constrain LLM-based alignment with stricter schema checks and lexicon-backed normalization. Calibrate merge-and-filter thresholds using development-set curves that track entity recall versus precision, supplemented by targeted manual audits on long or reordered sentences.

- **Reward shaping and training stability.** Remove or redesign attention-based rewards to improve generalization. Explore local, entity-centered structural rewards, for instance, $k$-hop dependency neighborhood similarity, as alternatives to global graph objectives. Stabilize GRPO training through pairwise or listwise advantage variants, and monitor optimization with KL divergence, clip fraction, and early stopping criteria keyed to both BLEU and entity metrics. Conduct finer-grained ablations to isolate the marginal contribution of each reward component.

- **Broader modeling and scalability.** Scale the RL framework to larger backbone models and multilingual settings to assess generalizability across languages and domains. Integrate lightweight domain adapters to enable efficient specialization for different terminology-sensitive fields (e.g., legal, engineering). Package the full pipeline, annotation tools, silver mining, RL training, and evaluation, for reproducibility and faster iteration.

- **Human evaluation and clinical validation.** Complement automatic metrics with targeted human evaluation on entity-rich phenomena such as negation, dosage expressions, and clinical qualifiers. Extend evaluation to additional medical subdomains (e.g., cardiology, pediatrics, radiology) to assess the framework's generalization beyond the entity categories and text types represented in MedEV.

# Closing Remarks

This thesis has demonstrated that multi-aspect reinforcement learning offers a principled and effective approach to improving entity-aware machine translation in terminology-sensitive domains. By jointly optimizing lexical fluency, entity fidelity, semantic consistency, and structural preservation within a unified GRPO framework, the proposed method achieves substantial gains in translation quality without sacrificing general-domain performance. The accompanying dataset resources and annotation framework provide a foundation for reproducible research in medical entity-aware translation. We hope that the insights from this work, on reward design, multi-objective optimization, and the interplay between entity accuracy and translation fluency, will inform future efforts toward building more reliable and clinically trustworthy machine translation systems.

# Publications


[1] Do Truong Dinh, **Trieu An Hoang**, Phi, Van-Thuy, Nguyen, Minh Le and Matsumoto, Yuji, *PolyMinder: A Support System for Entity Annotation and Relation Extraction in Polymer Science Documents. COLING*, 2025.

[2] Dinh-Truong Do, **Trieu Hoang An** , Van-Thuy Phi, Le-Minh Nguyen and Yuji Matsumoto. *Docora: A System for Interactive Knowledge Extraction and Visualization from Scientific PDFs. AAAI*, 2026.

[3] Phuong Nguyen, Cong Nguyen, Hiep Nguyen, Minh Nguyen, **An Trieu**, Dat Nguyen, Le-Minh Nguyen. *CAPTAIN at COLIEE 2024: Large Language Model for Legal Text Retrieval and Entailment. New Frontiers in Artificial Intelligence*, 2024.

[4] Chau Nguyen, Phuong Nguyen, Thanh Tran, Dat Nguyen, **An Trieu**, Tin Pham, Anh Dang, Le-Minh Nguyen. *CAPTAIN at COLIEE 2023: Efficient Methods for Legal Information Retrieval and Entailment Tasks. Workshop of the Tenth Competition on Legal Information Extraction/Entailment (COLIEE'2023) in the 19th International Conference on Artificial Intelligence and Law (ICAIL)*, 2023.

[5] **An Trieu**, Phuong Nguyen, Minh Le Nguyen *Enhancing Entity Aware Machine Translation with Multi-task Learning. SCIDOCA*, 2025.

[6] **Hoang-An Trieu**, Dinh-Truong Do, Chau Nguyen, Vu Tran, Minh Le Nguyen *Enhancing Document Retrieval in COVID-19 Research: Leveraging Large Language Models for Hidden Relation Extraction. SCIDOCA*, 2024.

[7] **Trieu An**, Long Nguyen, Minh Le Nguyen *Team LA at SCIDOCA shared task 2025: Citation Discovery via relation-based zero-shot retrieval. SCIDOCA*, 2025.

[8] Hai Nguyen, Hiep Nguyen, Trang Pham, Minh Nguyen, **An Trieu**, Dinh-Truong Do, Nguyen-Khang Le, Le-Minh Nguyen "JNLP at COLIEE 2025: Hybrid Large Language Model-based Framework for Legal Information Retrieval and Entailment", COLIEE 2025 in association with the 20th International Conference on Artificial Intelligence and Law, 2025.

[9] Nguyen, C., Luu, S., Tran, T., **Trieu, An**, Dang, A., Nguyen, D., Nguyen, H., Pham, T., Pham, T., Vo, T. T., Dol, D. T., Le, Nguyen-Khang, Nguyen, D. H., Le, N. C., Le, T. T., Bui, Q., Nguyen, P., Nguyen, H. T., Tran, V., & Nguyen, L. M."A Summary of the ALQAC 2023 Competition." In *15th International Conference on Knowledge and Systems Engineering (KSE)*, pp. 1–6, 2023.

[10] Dinh-Truong Do, Son T. Luu, Trang Pham, Trung Vo, Nguyen-Hoang Chu, Quang-Huy Chu, Cuong Nguyen, Minh Nguyen, **An Trieu**, Dat Nguyen, Thanh Tran, Cong Nguyen, Hiep Nguyen, Chau Nguyen, Nguyen-Khang Le, Dieu-Hien Nguyen, Binh Dang, Phuong Nguyen, Ha-Thanh Nguyen, Vu Tran, Le-Minh Nguyen. *A Summary of the ALQAC 2024 Competition. Proceedings of the 16th International Conference on Knowledge and System Engineering (KSE 2024)*, 2024.

# Awards

- **COLIEE 2023 Winning Team**: best performance on the Legal Case Retrieval Task 2 and Task 3.
- **COLIEE 2024 Winning Team**: best performance on the Legal Case Retrieval Task 4.
- **COLIEE 2025 Winning Team**: best performance on the Legal Case Retrieval Task 1 and Task 3.
- **SCIDOCA shared task 2025 Winning Team**: best performance on the SCIDOCA shared task 2025 Task 1.

# Bibliography


[1] Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aaron van den Oord, Alex Graves, and Koray Kavukcuoglu. Neural machine translation in linear time. *arXiv preprint arXiv:1610.10099*, 2016.

[2] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014.

[3] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[4] Jie Zhou, Ying Cao, Xuguang Wang, Peng Li, and Wei Xu. Deep recurrent models with fast-forward connections for neural machine translation. *Transactions of the Association for Computational Linguistics*, 4:371–383, 2016.

[5] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[6] Minh-Thang Luong and Christopher D Manning. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the 12th international workshop on spoken language translation: evaluation campaign*, pages 76–79, 2015.

[7] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

[8] Łukasz Kaiser and Samy Bengio. Can active memory replace attention? *Advances in Neural Information Processing Systems*, 29, 2016.

[9] Alex Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

[10] Denny Britz, Anna Goldie, Minh-Thang Luong, and Quoc Le. Massive exploration of neural machine translation architectures. *arXiv preprint arXiv:1703.03906*, 2017.

[11] Nal Kalchbrenner and Phil Blunsom. Recurrent continuous translation models. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1700–1709, 2013.

[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[13] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

[14] Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.

[15] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

[16] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937. PmLR, 2016.

[17] Po-Wei Chou, Daniel Maturana, and Sebastian Scherer. Improving stochastic policy gradients in continuous control with deep reinforcement learning using the beta distribution. In *International conference on machine learning*, pages 834–843. PMLR, 2017.

[18] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. Pmlr, 2018.

[19] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *International conference on machine learning*, pages 387–395. Pmlr, 2014.

[20] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.

[21] Shun-Ichi Amari. Natural gradient works efficiently in learning. *Neural computation*, 10(2):251–276, 1998.

[22] Sham M Kakade. A natural policy gradient. *Advances in neural information processing systems*, 14, 2001.

[23] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL `https://arxiv.org/abs/1707.06347`.

[24] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[25] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 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, 2002.

[26] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005.

[27] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.

[28] Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. From word embeddings to document distances. In *International conference on machine learning*, pages 957–966. PMLR, 2015.

[29] Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. Comet: A neural framework for mt evaluation. In *Proceedings of the 2020 conference on empirical methods in natural language processing (emnlp)*, pages 2685–2702, 2020.

[30] Pierre Baldi and Yosef Rinott. On normal approximations of distributions in terms of dependency graphs. *The Annals of Probability*, 17(4):1646–1650, 1989.

[31] Xinbo Gao, Bing Xiao, Dacheng Tao, and Xuelong Li. A survey of graph edit distance. *Pattern Analysis and applications*, 13(1):113–129, 2010.

[32] Nino Shervashidze, Pascal Schweitzer, Erik Jan Van Leeuwen, Kurt Mehlhorn, and Karsten M Borgwardt. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 12(9), 2011.

[33] Karsten M Borgwardt and Hans-Peter Kriegel. Shortest-path kernels on graphs. In *Fifth IEEE international conference on data mining (ICDM'05)*, pages 8–pp. IEEE, 2005.

[34] Maciej Modrzejewski, Miriam Exel, Bianka Buschbeck, Thanh-Le Ha, and Alex Waibel. Incorporating external annotation to improve named entity translation in nmt. In *Proceedings of the 22nd annual conference of the european association for machine translation*, pages 45–51, 2020.

[35] Shufang Xie, Yingce Xia, Lijun Wu, Yiqing Huang, Yang Fan, and Tao Qin. End-to-end entity-aware neural machine translation. *Machine Learning*, 111(3): 1181–1203, 2022.

[36] Zixin Zeng, Rui Wang, Yichong Leng, Junliang Guo, Xu Tan, Tao Qin, and Tieyan Liu. Extract and attend: Improving entity translation in neural machine translation. *arXiv preprint arXiv:2306.02242*, 2023.

[37] Sarthak Jauhari, Saisuresh Krishnakumaran, Dinkar Vasudevan, and Rahul Goel. Entity-aware joint translation of query and semantic parse. 2024.

[38] Yuchen Hu, Chen Chen, Chao-Han Huck Yang, Ruizhe Li, Dong Zhang, Zhehuai Chen, and Eng Siong Chng. Gentranslate: Large language models are generative multilingual speech and machine translators. *arXiv preprint arXiv:2402.06894*, 2024.

[39] Priyanka Sen, Alham Fikri Aji, and Amir Saffari. Mintaka: A complex, natural, and multilingual dataset for end-to-end question answering. In Nicoletta Calzolari, Chu-Ren Huang, Hansaem Kim, James Pustejovsky, Leo Wanner, Key-Sun Choi, Pum-Mo Ryu, Hsin-Hsi Chen, Lucia Donatelli, Heng Ji, Sadao Kurohashi, Patrizia Paggio, Nianwen Xue, Seokhwan Kim, Younggyun Hahm, Zhong He, Tony Kyungil Lee, Enrico Santus, Francis Bond, and Seung-Hoon Na, editors, *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1604–1619, Gyeongju, Republic of Korea, October 2022. International Committee on Computational Linguistics. URL https://aclanthology.org/2022.coling-1.138/.

[40] Simone Conia, Daniel Lee, Min Li, Umar Farooq Minhas, Saloni Potdar, and Yunyao Li. Towards cross-cultural machine translation with retrieval-augmented generation from multilingual knowledge graphs. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 16343–16360, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.914. URL https://aclanthology.org/2024.emnlp-main.914/.

[41] Simone Conia, Min Li, Roberto Navigli, and Saloni Potdar. SemEval-2025 task 2: Entity-aware machine translation. In *Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025)*. Association for Computational Linguistics, 2025.

[42] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

[43] Zi-Yi Dou and Graham Neubig. Word alignment by fine-tuning embeddings on parallel corpora. In *Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 2021.

[44] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1), January 2020. ISSN 1532-4435.

[45] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*, 2020.

[46] Nhu Vo, Dat Quoc Nguyen, Dung D. Le, Massimo Piccardi, and Wray Buntine. Improving Vietnamese-English medical machine translation. In Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue, editors, *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING*

*2024)*, pages 8955–8962, Torino, Italia, May 2024. ELRA and ICCL. URL https://aclanthology.org/2024.lrec-main.784/.

[47] National Library of Medicine. Health information in vietnamese (tiếng việt): Medlineplus multiple languages collection. https://medlineplus.gov/languages/vietnamese.html, 2025. Accessed: 15/03/2026.

[48] Peter MacCallum Cancer Centre. Multilingual cancer glossary - vietnamese. https://www.petermac.org/document/files/multilingual-cancer-glossary-vietnamese, 2023. Accessed: 15/03/2026.

[49] Bộ Y tế. Quyết định 1227/QĐ-BYT: Danh mục mã dùng chung đối với kỹ thuật, thuật ngữ chỉ số cận lâm sàng – Đợt 1. https://bvdkbaclieu.gov.vn/van-ban-phap-quy/quyet-dinh-1227-qd-byt-2025-danh-muc-ma-dung-chung-doi-voi-k.html, 2025. Truy cập: ngày 15/03/2026.

[50] Bộ Y tế. Quyết định 2010/QĐ-BYT: Ban hành tạm thời một số danh mục mã dùng chung phục vụ việc gửi dữ liệu chi phí khám bệnh, chữa bệnh BHYT. https://bvdkbaclieu.gov.vn/van-ban-phap-quy/quyet-dinh-2010-qd-byt-2025-ban-hanh-tam-thoi-mot-so-danh-mu.html, 2025. Truy cập: ngày 15/03/2026.

[51] Bộ Y tế. Quyết định 2427/QĐ-BYT: Ban hành Danh mục mã dùng chung thuật ngữ y học lâm sàng – Đợt 1. https://ttytbaolac.soytecaobang.gov.vn/thong-bao/quyet-dinh-2427-qd-byt-ngay-25-7-2025-ban-hanh-danh-muc-ma-dung-chung-thuat-ngu-y-hoc-lam-sang-d-1023603, 2025. Truy cập: ngày 15/03/2026.

[52] Bộ Y tế. Quyết định 2493/QĐ-BYT: Ban hành Danh mục mã dùng chung thuật ngữ y học lâm sàng – Đợt 2. https://syt.gialai.gov.vn/index.php/vi/laws/detail/Cong-van-so-1148-SYT-NVY-05-08-2025-V-v-trien-khai-Quyet-dinh-so-2493-QD-BYT-ngay-04-8-2025-cua-Bo-Y-te-ve-viec-ban-hanh-Danh-muc-ma-dung-chung-thuat-ngu-y-hoc-lam-sang-Dot-2-362/, 2025. Truy cập: ngày 15/03/2026.

[53] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45. Association for Computational Linguistics, 2020.

[54] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

[55] Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational*

*Linguistics: System Demonstrations*, pages 101–108. Association for Computational Linguistics, 2020.