COMP 390:
Literature Review

Janice Wong
jwong6@oxy.edu
Occidental College

1 Technical Background

1.1 The Foundation of Machine Translation

Machine translation (MT) refers to the use of computer algorithms to translate text or speech from one language to another. Over time, MT has evolved from rule-based and statistical methods to neural approaches. Early models relied on rule-based translation, which used manually defined grammar rules but struggled with linguistic ambiguity and the complexities of natural language. This led to Statistical Machine Translation (SMT), as introduced by Brown et al. (1993) [4], which applied probability models to predict the most likely equivalents for words and phrases. Eventually, SMT was outperformed by Neural Machine Translation (NMT), which was pioneered by Bahdanau, Cho, and Bengio (2015) [1], which employs neural networks to model the translation process. NMT systems can capture complex linguistic patterns and produce more fluent translations. Finally, the breakthrough came with Vaswani et al. (2017) [17], who proposed transformers, which utilizes self-attention mechanisms to process input data more efficiently, leading to improved translation quality.

1.2 Real-Time Translation and Latency Considerations

Standard transformer-based models are accurate, however they suffer from high inference latency, which is the time it takes for the model to generate a translation. This delay poses issues for real-time applications like live chat translations. Gu et al. (2018) [8] proposed Non-Autoregressive NMT (NAT), which reduces latency by generating entire sentences in parallel rather than sequential word prediction. Kasai et al. (2021) [10] demonstrated that using deep encoders and shallow decoders speeds up inference for autoregressive models without significant loss of quality.

1.3 Handling Slang and Informal Language

A major challenge in chat translation is processing informal language. Traditional NMT models are typically trained on structured text, so they often fail to translate slang, abbreviations, and code-switching accurately. Informal expressions and shortened forms, especially Internet slang, are highly context-dependent and evolve rapidly, making them difficult for models to interpret accurately.

2 Prior Work

2.1 Existing Solutions and Their Limitations

2.1.1 Commercial APIs

The most widely used real-time translation services, such as Google Translate [18] and DeepL [5], rely on high-performing NMT models trained on massive datasets. While effective for formal text, they often struggle with slang, abbreviations, and informal language [2]. Additionally, these proprietary APIs lack overall transparency, which makes customization difficult for the user.

2.1.2 Open-Source NMT Models

There are some open-source alternatives, such as Marian NMT [9] and OPUS-MT [16], which allow for customization. However, fine-tuning these models requires extensive domain-specific data. This type of data is often unavailable for informal speech and chat conversations.

2.2 A New Approach is Needed

As mentioned already, existing solutions provide high-quality translations for formal contexts, but struggle with real-time informal conversations. Unlike prior work, this project aims to:

- Reduce latency: Implementing a shallow decoder structure [10] speeds up translation.
- Improve slang handling: Fine-tuning transformers for slang and abbreviation recognition is essential for accurate translation of informal language.
- Support code-switching: Adapting multilingual models to handle mid-sentence language changes can improve translation accuracy in multilingual conversations.

So, this system will improve standard APIs with these strategies and make real-time chat translation more effective in casual communication settings.

3 Methods

For this project, I will implement a real-time chat translation system optimized for casual messaging which includes handling slang and informal speech. Given the challenges identified earlier, a transformer-based neural machine translation (NMT) model will be used due to its superior performance in multilingual translation tasks [17]. Specifically, the M2M-100 model [7] will serve as the base model, as it supports direct translation between 100+ language pairs without relying on English as an intermediary.

Additionally, a slang-detection and preprocessing module will be implemented using tBERT [12] to classify informal expressions and replace them with their standardized equivalents before translation. This pre-processing step ensures that the translation mode is fed with more structured input. This improves translation quality for slang-heavy messages.

And to address real-time constraints, I will employ non-autoregressive translation (NAT) techniques [8] to reduce latency. Traditional autoregressive transformers generate one token at a time, but NAT models generate whole sentences in parallel, which significantly improves speed.

3.1 Hyperparameters and Exploration

Several hyperparameters will be explored and optimized to balance translation speed and accuracy.

1. Model Size and Quantization
   (a) Base model: The standard M2M-100 model has 15 billion parameters, but for real-time applications, I will explore smaller versions.
   (b) Quantization: I will apply 8-bit quantization [19] to reduce computational overhead while maintaining translation quality.
2. Beam Search vs. Greedy Decoding Beam Search Width (k):
   Beam search improves translation quality by considering possible translations at each step. I will compare k = 1 (fastest but less accurate) and k = 5 (higher quality but slower) to determine the best trade-off for real-time translation.
3. Slang Handling Preprocessing
   (a) Threshold for slang replacement: The slang-detection model will output a confidence score. I will experiment with different confidence thresholds (e.g., 0.7 vs. 0.9) to balance between preserving informal expressions and ensuring comprehensibility.
   (b) Dynamic slang lexicon updates: To ensure adaptability, the slang lexicon will be updated dynamically using data collected from real-world chat logs. Frequency-based updating strategies will be tested.
4. NAT Mask-Prediction Iterations (n):
   NAT models use a masked-prediction approach. In this approach, the model initially generates a rough translation where some words may be missing or incorrect. The model then refines the translation with subsequence iterations by masking uncertain words and predicting them again based on the surrounding context. I will experiment with n = 1 iterations (fastest but less refined) vs. n = 3 (higher accuracy but more compute-intensive).

Shallow Decoder Optimization: Kasai et al. (2021) [10] demonstrated that deep encoders and shallow decoders improve inference speed without degrading quality. I will test reducing decoder depth from 6 layers to 3 layers.

3.2 Implementation Tools

The project will be implemented using PyTorch [14] for model training and inference. I will use Hugging Face's Transformers library for integrating M2M-100 and tBERT [6]. Deployment will be optimized using ONNX Runtime to accelerate inference on CPU/GPU [15]. Real-time translation will be enabled via WebSockets for low-latency message delivery.

4 Evaluation Metrics

The effectiveness of this real-time chat translation system can be evaluated through a combination of traditional machine translation (MT) metrics, real-time performance benchmarks, and user evaluations. Prior work in NMT has largely relied on standard benchmarks such as BLEU, METEOR, and chrF, but these metrics may not fully capture the nuances of slang translation and real-time constraints. Therefore, along with these standard metrics, this project will incorporate latency measurements, user comprehension testing, and a slang translation accuracy metric.

4.1 Standard Machine Translation Metrics

Existing translation research primarily evaluates models based on metrics that are accuracy-driven:

1. BLEU (Bilingual Evaluation Understudy) Score [11]

Measures n-gram overlap between the model output and reference translations. BLEU is widely used in MT but struggles with evaluating informal or creative language structures.

2. METEOR [3]
   An improvement over BLEU, incorporating synonym matching and stemming to better assess semantic similarity. This is useful for informal text where slang may have multiple valid equivalents.
3. chrF [13]
   A character-level F-score metric, which is useful for languages with complex word structures (e.g., German, Finnish) and short informal texts (e.g., chats, tweets).

Prior research suggests that for an NMT model to be considered competitive, it should ideally achieve at least 30-40 BLEU on general-domain translation tasks [18] and [17]. However, informal text often lacks standardized reference translations. This makes BLEU less reliable.

4.2 Real-Time Performance Metrics

Speed is obviously a critical evaluation factor since the system is designed for chat translation. The following latency benchmarks will be used:

1. Translation Latency
   The time taken for a message to be translated, measured in milliseconds (ms). For human perception, a response time of around 100 milliseconds is perceived as instantaneous. Therefore, I will aim to achieve a latency in the approximate 100-200 milliseconds threshold.
2. Throughput
   Number of words translated per second, to assess the model's ability to scale with real-time usage.

To optimize latency, this project will explore quantization [19] and non-autoregressive decoding [10], which have been shown to improve speed without significant accuracy loss.

4.3 Slang-Specific Accuracy Metrics

Since traditional MT metrics do not capture slang translation quality, I will develop a specialized evaluation approach.

1. Slang Translation Accuracy
   A manually curated benchmark set of slang expressions and informal phrases will be used to evaluate performance. Accuracy will be measured as the proportion of correctly translated slang terms based on human annotators' judgments. A score above 90% is considered competitive for specialized translation tasks.
2. Code-Switching Handling
   Since real-world conversations often mix languages, evaluating code-switching translations separately may be needed. Performance will be assessed based on how accurately it translates mixed-language phrases.

4.4 Human Evaluation and User Feedback

Automated metrics do not always align with a human's perception of translation quality, especially for informal language. To address this, I will facilitate a small-scale user study where bilingual participants rate translations based on these two factors:

1. Fluency and naturalness
   How well the translation reads as natural conversation. Rated on a 1-5 Likert scale.
2. Comprehensibility
   Whether the meaning of the translated text is clear and understandable to a native speaker of the target language. Rated on a 1-5 Likert scale.

A successful model should achieve at least a 4.0/5 on fluency and comprehensibility in human evaluations.

References

[1] Bahdanau, Dzmitry, Cho, KyungHyun, and Bengio, Yoshua. "Neural Machine Translation by Jointly Learning to Align and Translate". In: *Proceedings of the International Conference on Learning Representations*. 2015.

[2] Baldwin, Timothy et al. "How Noisy Social Media Text, How Diffrnt Social Media Sources". In: *Proceedings of the Sixth International Joint Conference on Natural Language Processing*. 2013.

[3] Banerjee, Satanjeev and Lavie, Alon. "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*. 2005.

[4] Brown, Peter F. et al. "The Mathematics of Statistical Machine Translation: Parameter Estimation". In: *Computational Linguistics* (1993).

[5] DeepL. URL: `https://developers.deepl.com/docs`.

[6] Face, Hugging. URL: `https://huggingface.co/docs/transformers/en/index`.

[7] Fan, Angela et al. “Beyond english-centric multilingual machine translation”. In: *The Journal of Machine Learning Research* 22 (1 2021).

[8] Gu, Jiatao et al. “Non-Autoregressive Neural Machine Translation”. In: *Proceedings of the International Conference on Learning Representations*. 2018.

[9] Junczys-Dowmunt, Marcin et al. “Marian: Fast Neural Machine Translation in C++”. In: *Proceedings of ACL 2018, System Demonstrations*. 2018.

[10] Kasai, Jungo et al. “Deep Encoder, Shallow Decoder: Reevaluating Non-Autoregressive Machine Translation”. In: *Proceedings of the International Conference on Learning Representations*. 2021.

[11] Papineni, Kishore et al. “BLEU: a method for automatic evaluation of machine translation”. In: *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. 2002.

[12] Peinelt, Nicole, Nguyen, Dong, and Liakata, Maria. “tBERT: Topic Models and BERT Joining Forces for Semantic Similarity Detection”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 2020.

[13] Popović, Maja. “chrF: character n-gram F-score for automatic MT evaluation”. In: *Proceedings of the Tenth Workshop on Statistical Machine Translation*. 2015.

[14] PyTorch. URL: `https://pytorch.org`.

[15] Runtime, ONNX. URL: `https://onnxruntime.ai`.

[16] Tiedemann, Jörg and Thottingal, Santhosh. “OPUS-MT - Building open translation services for the World”. In: *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*. 2020.

[17] Vaswani, Ashish et al. “Attention Is All You Need”. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 2017.

[18] Wu, Yonghui et al. *Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation*. 2016.

[19] Zafrir, Ofir et al. “Q8BERT: Quantized 8Bit BERT”. In: *5th Workshop on Energy Efficient Machine Learning and Cognitive Computing*. 2019.