Babelbit White Paper

LLM-based Utterance Completion for Low-Latency Machine Translation

Author: Matthew Karas, founder of Babelbit Ltd
Date: September 2025
Website: babelbit.ai · **X:** @babelbit · **Email:** info@babelbit.ai

1 Introduction

Latency is the key linguistic barrier to usable real-time speech translation. The other major barriers are to do with speech encoding and vocoding. Even though text-based machine translation can run many times faster than real time, conventional simultaneous translation must often wait for clause-completing information (e.g., sentence-final verbs), creating delays that disrupt turn-taking and natural flow. Babelbit reframes this as a **probabilistic adequacy** problem: we propose that it is possible to commit to a translation as soon as the utterance can be *adequately predicted* to keep the conversation going.

This white paper provides explanations throughout both for lay readers and machine learning experts. It defines the meaning and usage of our adequacy-based metrics (EATP, Lead, ACS, EA_γ), and outlines a planned **LLM-based judge** for human-aligned adequacy scoring, and presents a possible **two-stream architecture** (low-latency vs accuracy). We also reference recent work, indicating that **LLMs are central to a new strand of machine translation research** (e.g. Cambridge University's *SimulS2S-LLM*, and Kyutai's *Hibiki*).

2 Background: Why Latency Persists — and Why LLMs Help

Description

Traditional MT either buffers entire sentences or proceeds cautiously, phrase-by-phrase, in simultaneous mode. In languages with delayed disambiguation, reliable translation has historically meant **waiting**. LLMs change this. Because they are trained for **global coherence**, they can predict **utterance completions** that discern meaning before the final words arrive.

Detail

Let the source utterance be
$X = (x_1, \ldots, x_N)$.

Classical next-token models optimize
$P(x_{k+1} \mid X_{1:k})$.

For latency, we care about **global continuation**:
$P(X_{k+1:N} \mid X_{1:k})$.

After each prefix $X_{1:k}$ we propose a set of full candidates $\mathcal{C}_k$ with confidences $q_k(c)$ and assess **adequacy**
$A(c, X)$.

3 Core Concepts

3.1 Utterance Completion vs Next-Word Prediction

Description

Next-word prediction: local semantic plausibility at token level, which is a key feature of all LLMs.

Utterance completion: global semantic adequacy for the *remainder* of the utterance, given the incrementally growing prefix, enabling earlier translation commitments.

Semantic vs Lexical Accuracy: While we will be scoring lexical accuracy of predictions, it should be noted that a fully-accurate lexical prediction, will always be semantically accurate, but a single small lexical error can completely change the meaning. So in practice, it is likely that we will

optimise for semantic similarity. This is consistent with the strengths of LLMs.

Detail

We evaluate candidates $c \in \mathcal{C}_k$ using an adequacy function combining lexical and semantic components:

$$A(c, X) = \alpha\, L(c, X) + (1 - \alpha)\, S(c, X), \quad \alpha \in [0, 1].$$

The **Earliest Adequate Translation Point (EATP)** is the first prefix where adequacy clears a threshold τ:

$$k^* = \min\{\, k \in \{1, \ldots, N\} : \ \exists\, c \in \mathcal{C}_k,\ A(c, X) \geq \tau \,\}.$$

3.2 Our Metrics for Early Adequacy

Accuracy is only useful, if an accurate prediction can be made significantly early, so we have devised metrics as follows:

Recognition Lead (normalized earliness):

$$\mathrm{Lead}_\tau = \frac{N - k^*}{N - 1}.$$

Adequate Commitment Score (ACS) (earliness × confidence × adequacy margin):

$$\mathrm{ACS}_\tau = \frac{N - k^*}{N - 1} \cdot p^* \cdot \frac{A(c^*, X) - \tau}{1 - \tau}.$$

Early Adequacy Area (EA_\gamma) (threshold-free, early-weighted):

$$\mathrm{EA}_\gamma = (1 - \gamma) \sum_{k=1}^{N} \gamma^{k-1} \sum_{c \in \mathcal{C}_k} q_k(c)\, A(c, X).$$

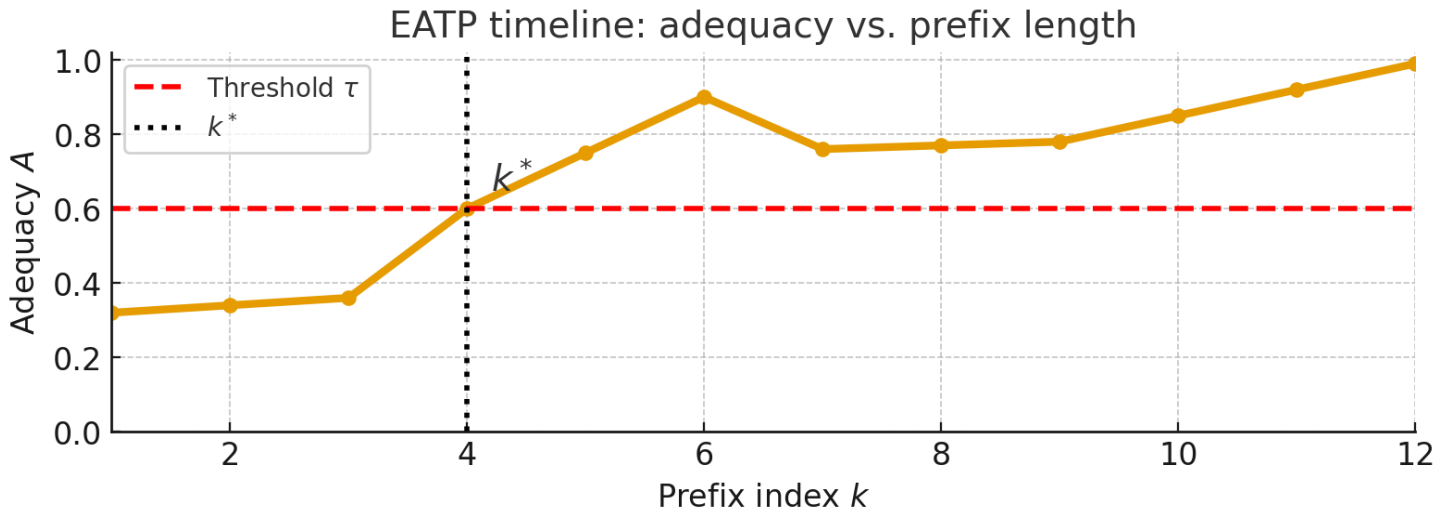

Figure 1. EATP timeline (adequacy vs prefix)
This schematic EATP timeline illustrates a typical progression: adequacy rises rapidly and crosses the threshold, τ, early, i.e. where it could be used for a translation. It then plateaus while semantic gains are minimal, and then increases again once the prefix contains almost the entire utterance.

4 Planned Development — LLM-based Adequacy Judge

Background

Embedding metrics for semantic similarity (e.g., chrF, BERTScore, COMET) are useful to kick-off the scoring process for iterative improvement but may diverge from human judgments of similarity in real dialogue. We are developing an **LLM-based judge** J_θ which scores conversational adequacy in a more natural way, and returns both a **category** and a **calibrated** score $[0, 1]$. Being less obviously deterministic, it is also much harder to game, so contributors will not be able to submit iterations which score highly, but have

Outline of Approach

The judge can run **reference-free** (source ↔ candidate) or **reference-based** (with a trusted translation). We fit a monotone calibration g (e.g., isotonic regression) so raw outputs align with empirical adequacy; pairwise training (Bradley–Terry / TrueSkill) can improve stability. Versioning preserves comparability as prompts/configs evolve.

5 Two-Stream Architecture

Description

We operate **two streams in parallel**:

1. **Low-latency stream** — emits speech as soon as adequacy is met (driven by utterance completion).
2. **High-accuracy stream** — produces a trustworthy **translation of record** with **zero predictive shortcuts**.

Detail

The low-latency path uses short decision horizons and calibrated confidence; the high-accuracy path can translate with full context (optionally conditioning on the low-latency output as a prior). The design is robust: later segments can revise the record without disrupting live speech.

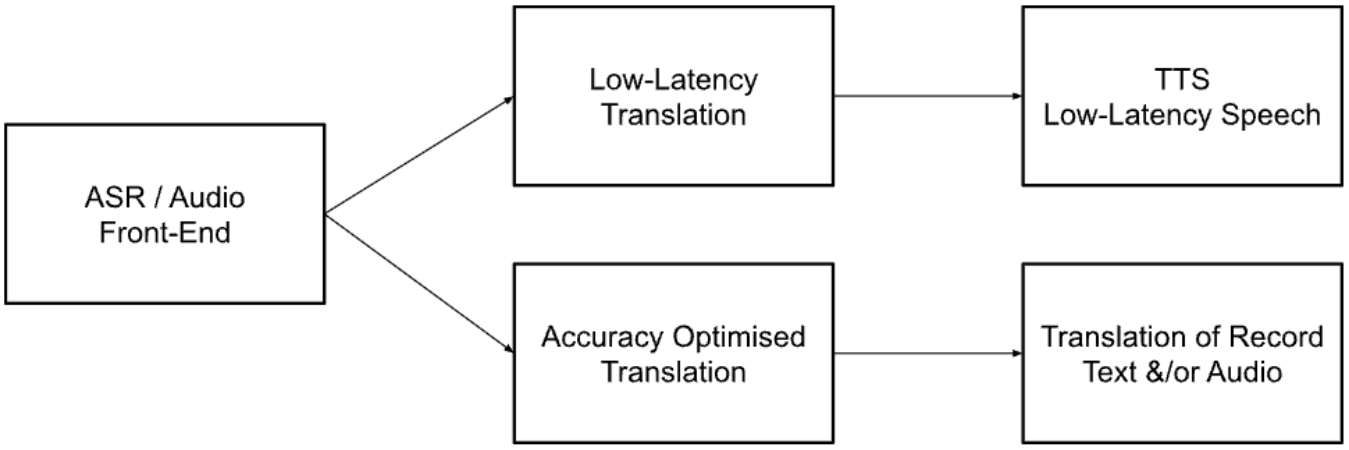

Figure 2. Two-stream architecture

6 Confidence Calibration and Decision Policies

Description

Early commitment requires **trustworthy confidence**. We map raw scores to a probability of adequacy $\hat{P}_k$ via monotone calibration and **commit at the earliest** k with $\hat{P}_k \geq q$ (target quality). While we are training the

prediction engine to make better and better predictions, we do not need the engine to have any internal representation of which step is good enough. However, to deploy the engine in practice, the latency gain can only be made by having an accurate representation or **confidence quotient**. This can be applied differently by context, e.g. a complex or subtle legal point will need a higher threshold of semantic similarity than small talk. LLMs are ideally suited to making such judgements on the fly.

Detail

We log $(p_k, \text{features}, y_k)$ where $y_k = \mathbf{1}\{A(\hat{c}_k, X) \geq \tau\}$. Calibration options include per-k **isotonic regression** and feature-based **logistic/gradient** models constrained to be monotone in p_k. Conformal adjustments can control risk at level α; hysteresis avoids oscillation around thresholds.

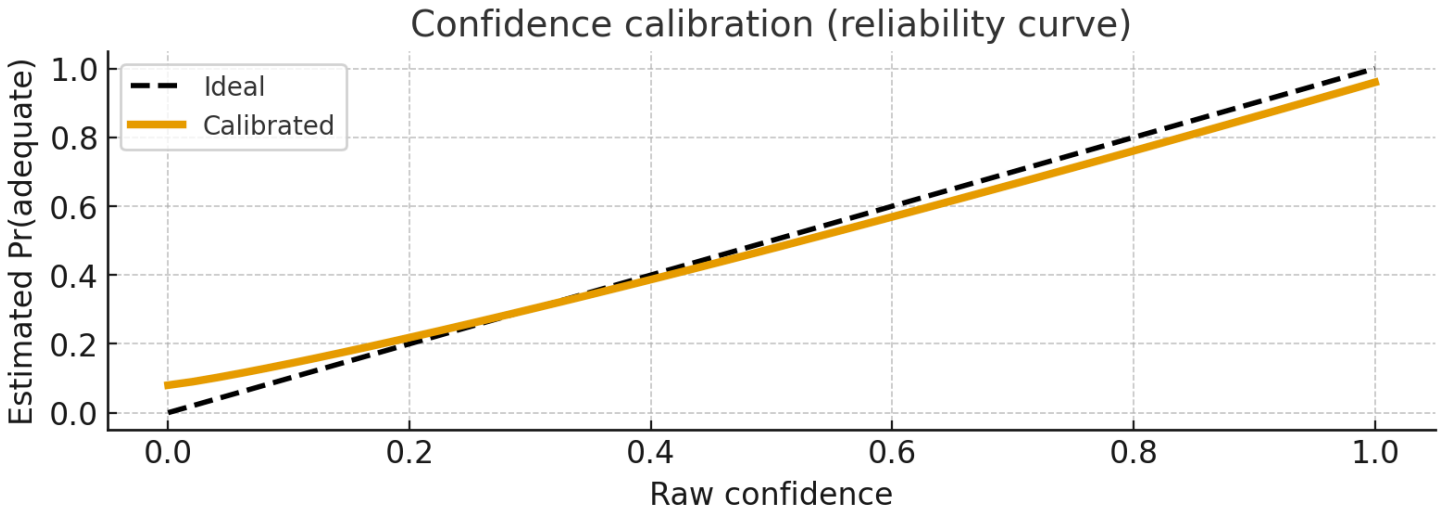

Figure 3: Confidence Calibration

7 Worked Example and Contributor-Driven Improvement

Consider the utterance **“Hello Tony, how is it going at your end?”** with $N = 9$.

With $\tau = 0.60$ and $\alpha = 0.3$, EATP occurs at $k^* = 2$.

k	Candidate ĉ_k	L(c,X)	S/J(c,X)	A(c,X)	Adequate (A≥τ)?
1	*Hello …*	0.30	0.45	0.40	No

k	Candidate ĉ_k	L(c,X)	S/J(c,X)	A(c,X)	Adequate (A≥τ)?
2	*Hello Tony, how are you*	0.40	0.65	0.605	**Yes (k*=2)**
3	*Hello Tony, how are things going*	0.55	0.70	0.685	Yes
4	*Hello Tony, how is it going*	0.62	0.74	0.712	Yes

Lead: $\mathrm{Lead}_\tau = (N - k^*)/(N-1) = 0.875.$
Assume $p^* = 0.30$. **ACS:**
$\mathrm{ACS}_\tau \approx 0.875 \times 0.30 \times \frac{0.605 - 0.60}{0.40} \approx 0.0033.$

Incentivised iteration. Contributors submit improved predictors, calibrators, and selection
policies. Each round is scored on **Lead**, **EA_γ**, and **ACS**. Over iterations, contributors
tend to (i) reduce k^*, (ii) increase calibrated $\hat{P}_k$, and (iii) widen adequacy margins —
raising composite scores.

8 Deployment Model

Babelbit will be offered as **SaaS** for meetings and collaboration, plus **on-premises licensing** for security-sensitive sectors. The two-stream output supports both **live interaction** and a trusted **translation of record**. Having developed software using an incentivised network of contributors, across the Bittensor ecosystem, we will support that ecosystem by using other decentralised services, e.g. Chutes, Hippius, Macrocosmos Gravity etc.

9 Possible Future Challenges

- **Live text-to-text translation** (e.g. real-time multilingual subtitling).
- **Speech-in, text-out** to optimize a novel low-latency LLM audio-ingest architecture which is under development.

- **One-shot speech-to-speech** models that bypass intermediate text.

10 Selected References

- Woodland et al., *SimulS2S-LLM: Unlocking Simultaneous Inference of Speech LLMs for Speech-to-Speech Translation* (2025).
- Kyutai Research, *Hibiki* project reports (2024–2025).
- Popović, *chrF: character n-gram F-score for automatic MT evaluation* (2015).
- Zhang et al., *BERTScore: Evaluating Text Generation with BERT* (2020).
- Rei et al., *COMET: A Neural Framework for MT Evaluation* (2022).

11. Conclusion

LLM-based utterance completion provides a principled route to **lower latency** in speech translation. By formalizing adequacy, calibrating confidence, and using a two-stream design, the system maintains conversational flow while safeguarding the state-of-the-art in accuracy. A contributor-incentive loop steadily improves performance, offering a credible path from prototype to deployment.